NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

DRAG OPTIMIZATION OF LIGHT TRUCKS USING COMPUTATIONAL FLUID DYNAMICS

by

Nathan A. Williams

September 2003

Thesis Advisor: Joshua H. Gordis
Thesis Co-Advisor: Dan Boger

Approved for public release; Distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Sep 2003	Master's Thesis

4. TITLE AND SUBTITLE: Title (Mix case letters)	5. FUNDING NUMBERS
6. AUTHOR(S)	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; Distribution is unlimited	

13. ABSTRACT *(maximum 200 words)*

There are 80 million light trucks on the road today with suboptimal aerodynamic forms. Previous research has found that several miles per gallon can be saved by specifically tailoring truck bodies for reduced aerodynamic drag. Even greater savings can be obtained if the shape of the trucks is numerically optimized. This could reduce fuel consumption in the United States by billions of gallons per year.

The purpose of this research is to develop and quantify optimal light truck canopy designs using computational fluid dynamics (CFD). Both two-dimensional and three-dimensional models are used to do this. Initially, this research focuses on quantifying and generalizing the effects of traditional automotive aerodynamic accessories, such as canopies and air dams. Once the effects of various form factors are quantified an optimization of the canopy is performed. This thesis demonstrates a method for drag reduction using CFD and traditional numerical optimization techniques. Lastly, the optimized forms are physically constructed and their effects on fuel economy are compared to the CFD prediction.

The results indicate that the CFD formulation provides an accurate predictor for improving fuel economy and drag characteristics. The prototype air dam and optimally shaped canopy generated a 21.23% savings in terms of fuel economy.

14. SUBJECT TERMS	15. NUMBER OF PAGES
shape optimization, CFD, computational fluid dynamics, drag, canopy, air dam, light truck, mpg, miles per gallon, fuel efficiency, fuel economy	185
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for Public release; Distribution is unlimited

DRAG OPTIMIZATION OF LIGHT TRUCKS USING COMPUTATIONAL FLUID DYNAMICS

Nathan A. Williams
Lieutenant, Civil Engineer Corps, United States Naval Reserve
B.S., United States Naval Academy, 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2003

Author: Nathan A. Williams

Approved by: Joshua H. Gordis, PhD
Thesis Advisor

Dan Boger, PhD
Thesis Advisor

Anthony J. Healey
Chairman, Department of Mechanical Engineering

Dan Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

There are 80 million light trucks on the road today with suboptimal aerodynamic forms. Previous research has found that several miles per gallon can be saved by specifically tailoring truck bodies for reduced aerodynamic drag. Even greater savings can be obtained if the shape of the trucks is numerically optimized. This could reduce fuel consumption in the United States by billions of gallons per year.

The purpose of this research is to develop and quantify optimal light truck canopy designs using computational fluid dynamics (CFD). Both two-dimensional and three-dimensional models are used to do this. Initially, this research focuses on quantifying and generalizing the effects of traditional automotive aerodynamic accessories, such as canopies and air dams. Once the effects of various form factors are quantified an optimization of the canopy is performed. This thesis demonstrates a method for drag reduction using CFD and traditional numerical optimization techniques. Lastly, the optimized forms are physically constructed and their effects on fuel economy are compared to the CFD prediction.

The results indicate that the CFD formulation provides an accurate predictor for improving fuel economy and drag characteristics. The prototype air dam and optimally shaped canopy generated a 21.23% savings in terms of fuel economy. This was an improvement for the test truck from 19 mpg to 23.17 mpg. Ultimately, this thesis demonstrates a practical example of geometric shape optimization and validates the results of the optimization with a full scale test.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to thank Mr. Mike Leatherwood and the US Army Tank-Automotive and Armaments Command for supporting this thesis. Additionally, the author would like to thank Professor Joshua Gordis for his support, insight and guidance in the development of this thesis. Most importantly, the author would like to thank his wife, Holly, and his family for their incredible patience and support during the last two years.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

To date there has been very little published research into the aerodynamic characteristics of light trucks. Most ground vehicle aerodynamic research has been performed on race vehicles, consumer automobiles (sedans, minivans, sport utility vehicles) and large class A, semi-tractor trailers. In addition, almost all of the research on consumer grade vehicles has been performed by the large auto makers in Detroit and Japan for their commercial use. These results, understandably, are not available for use by the institutional researcher.

In previous years, it was only practical for large automakers to perform aerodynamic experiments on cars and trucks because of the expense associated with operating a wind tunnel. Today, computational fluid dynamics (CFD) programs make it possible for institutional research organizations, with limited funding, to perform meaningful research. In addition, problems that would normally be discarded because of excessive cost/benefit ratios can now be explored by the interested researcher. Lastly, iterative shape optimization is impractical in a wind tunnel due to the costs of setup and number of runs required for a successful optimization. These costs would be exorbitant even for the largest automotive manufacturers. Fortunately, the costs of running a CFD optimization are measured more in time and CPU cycles than the direct costs of setting up wind tunnel experiments. This thesis will approach the optimization of an under-researched field, light truck aerodynamics, using a CFD based approach

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. MOTIVATION

There are over 80 million light trucks on the road in America today. The federal government owns at least 1 million of these vehicles for mundane use and has interest in using commercially available light trucks to supplant the aging population of MILSPEC trucks and tactical vehicles. For example, ¾ ton and 1 ton trucks from Daimler-Chrysler, and Ford are being considered as replacements for the High Mobility Multipurpose Wheeled Vehicle (HMMWV).

Current truck designs exhibit suboptimal fuel economy and tactical performance (speed/range) characteristics. Preliminary research suggests that drag coefficients for light trucks can be reduced from .5 to .3 [Ref. 1]. Reductions in drag of this order have been shown to improve fuel economy by several miles per gallon on average. If all trucks were to improve their drag coefficients by this margin, the United States would realize billions of gallons of fuel savings per year. As well, a tactical vehicle with improved aerodynamics would have an extended range, require less support, and be able to operate at higher speeds.

B. OBJECTIVE

The objective of this thesis is to conduct a thorough analysis of current light-truck aerodynamics through computer modeling and provide recommendations for improvement. The primary tool that will be used to accomplish this will be computational fluid dynamics (CFD). This thesis will quantify the effects of aerodynamic accessories on light-truck aerodynamics through CFD modeling. Once general effects of accessories (such as canopies) have been quantified, the accessory shape will be optimized. A large portion of this thesis will demonstrate how a shape optimization is possible using current systems. Lastly, the optimized model will be physically constructed and attached to a test vehicle to validate the CFD models.

C. HISTORY OF LIGHT TRUCKS

Light duty trucks have been around almost since the advent of the automobile. In fact, there was a Model-T derivative that was sold as a truck in 1918. It has just been in recent years that the light duty truck has gained a market share that nearly rivals that of automobiles in the United States. In 1968 there were roughly 83 million cars and 17 million trucks. In the year 2000, there are 133 million cars and 88 million trucks. Light truck growth has outpaced cars by a huge margin over the 32 year period: 417% vs. 60%. [Ref. 2].

This trend has not translated to the level of effort placed on improving light truck aerodynamics though many improvements have been made from the initial Model-T of 1918. A look at the history of Ford trucks is instructive:

Figure 1. Model-T from 1918

The first 40 years of development brought covered wheels and curved fore-features to the Ford line.

Figure 2. F-100 from 1951

The new F-100 introduced in 1965 almost appears to be a step backward in terms of aerodynamics. It is boxier and arguably less aerodynamic but provides the consumer with greater capacity in terms of payload and towing.

Figure 3. F-100 from 1965

The 1973 model, again, almost seems a step backward in terms of aerodynamics but is actually better because of a more reclined windshield.

Figure 4. F-150 from 1973

The early 90's saw small incremental improvements with a slightly more streamlined shape.

Figure 5. F-150 from 1991

The 1997 F-150 from ford is proclaimed (by all automotive journalists) to be the most aerodynamic light truck form to date, which should be obvious to the casual observer based upon its almost car-like curves. Ironically, the curved shape is cited as one of the reasons that Ford's newest design lost market share, due to consumer preference for "tough" looking trucks.

Figure 6. F-150 from 1997

Typical modern trucks have a coefficient of drag of around .5 while their predecessors wallowed in the .8 to 1.0 range.

D. PREVIOUSLY CONDUCTED RESEARCH

Almost all publicly available ground vehicle aerodynamics research has focused on cars, semi trailers, airplanes, and race cars. All of these vehicle categories have large geometric differences with light trucks in terms of size and configuration. As a result, the data gleaned from these studies is not necessarily relevant to light trucks. A significant level of privately held research, by the major auto manufacturers, has focused on light trucks but not to the extent of which has focused on cars. This is very likely due to the market forces at work and consumer preferences. Drivers of trucks have traditionally been far more interested in payload, ground clearance, and towing capacity. These factors do not help with fuel economy and automakers have had little incentive to do otherwise. The attitudes of automakers may change with the recent legislation passed by Congress:

Under the rule, which NHTSA proposed in December and adopted after a public comment period, the current standard of 20.7 miles per gallon will increase

to 21 miles per gallon in model year 2005, 21.6 miles per gallon in model year 2006 and 22.2 miles per gallon in model year 2007. The current standard for passenger cars is 27.5 miles per gallon [Ref. 3]

The exception to the general lack of research on light trucks is found in 1999 publication from NASA's Dryden Flight Research Center in Edwards, CA [Ref. 4]. Though the paper was published in 1999, most of the research was performed between 1973 and 1982 in response to the "oil crisis". The research was performed on a full scale van that was roughly the size of a light duty truck. The test vehicle that NASA used is shown below:

Figure 7. NASA Dryden Research Vehicle – Box Shape

NASA's interest was primarily directed at improving semi trailer performance but these results are even more appropriate for light trucks. They found that a practical drag coefficient (Cd) of .25 was possible when a rounded front was implemented along with a truncated "boattail" shaped back. See figure 8 on the next page:

Figure 8. NASA Vehicle with curved front and boat tail

The NASA research provided much insight into where the preliminary CFD investigations should begin for this thesis. They investigated the effects of having a number of accessories attached to the van. The accessories can be defined as follows:

1. Rounded front – shape similar to the front of a modern blunt nose bus (See the front end of the vehicle above).
2. Rounded rear – shape similar to the rear of a modern bus
3. Sealed underbody – this modification entails covering the axles from front to rear with streamlined surfaces.
4. Boat tail rear – this entails tapering the rear of the van to a point that extends the van by a great margin.
5. Truncated boat tail – this shape is a greatly shortened version of the above boat tail (See the tail end of the vehicle above).

Key conclusions that were gleaned from the NASA publication are:

6. A square frontal section and rear section are the worst possibility.
7. Rounding of the front is most beneficial if it is done both horizontally and vertically.

8. Even better than rounding at the rear of the vehicle is to add a “boat tail” that lengthens the vehicle and tapers both horizontally and vertically.
9. A sealed underbody is preferable to exposed axles and frame by a significant margin.
10. A truncated boat tail is nearly as aerodynamic as a full boat tail (.25 Cd vs. .24 Cd).

E. CONTENTS

The next two chapters include the problem formulation and the results. The traditional understanding of drag, CFD problem formulation, and the development of the CFD optimization method is included in Chapter III. The problem formulation methods are used to develop the results in Chapter IV. Chapter IV includes CFD formulations of two dimensional and three dimensional models of a ¾ ton truck. These models are optimized using a conjugate direction method in the last half of Chapter IV. Chapter V includes the methods the used for constructing the physical models and the results that were recorded upon testing their effect on fuel economy. Conclusions and comparisons of the physical model are provided in Chapter VI. Lastly, Chapter VII offers some recommendations for future research.

III. PROBLEM FORMULATION

A. AERODYNAMIC DRAG ON VEHICLES

Aerodynamic drag is a surprisingly complicated subject even though casual observers feel that they have a notion of its origin. Aerodynamic drag is actually a subcategory of aerodynamic forces. The forces include lift, drag and side forces. The three forces can be added as vectors and considered the total aerodynamic force. When the total aerodynamic force acts at a location other than the center of gravity, moments can be attributed to the aerodynamic force. That is, when the center of pressure (application point of aerodynamic force) is not at the center of gravity, the object will experience moments due to the force. In traditional aircraft terms these moments are know as yawing, rolling and pitching moments [Ref. 1].

Ground vehicles are primarily concerned with drag which is the component of the aerodynamic force that acts in the direction of the free stream velocity. There are two components to drag: friction drag and pressure drag. Friction drag or skin friction is attributed to the interaction between the solid molecules of the vehicle and the molecules of the air. Friction drag is minor in comparison to pressure drag. Most reputable sources claim that friction drag, at speeds generally exhibited by ground vehicles, only accounts for 10% of total drag or less [Ref. 5]. These claims are consistent with the CFD simulations performed for this thesis.

Pressure drag is clearly the source of drag that is critical to understand as it makes up 90% of drag. It is also the most difficult to understand as it is complicated by turbulence and the detachment of the boundary layer in aerodynamic problems. The simplest ways to conceptualize drag is to analyze the air particles in terms of conservation of linear momentum and Bernoulli's equation.

$$F = m\frac{\Delta V}{\Delta t}$$

$$\frac{P}{\rho} + \frac{1}{2} V^2 + g * Z = C$$

$m = mass$

$t = time$

$P = \Pr essure$

$\rho = density$

$V = Velocity$

$g = gravitational\ acceleration$

$z = height$

Clearly as a vehicle approaches an air particle that has no momentum, a force has to be applied to give the particle enough velocity for it to move aside. The vehicle forces the particle aside and often imparts the particle with some momentum that continues after the vehicle has passed. The reactionary force of all relevant particles is the drag on the vehicle. An even better way to conceptualize drag is through Bernoulli's equation. As air flows around a body, the velocity and pressure change locally. For example, frontal regions become stagnant areas with low velocities and high pressure. This is evidenced by the pressure domes seen in many experiments. The pressure is integrated around the body and the portion directed in the free stream direction is considered the drag force [Ref. 1].

As mentioned previously, separation of the boundary layer and the ensuing turbulence complicates the problem dramatically. The turbulence to the rear of the body actually reduces the pressure further than the free stream creating a suction effect on the body. In White 1999 [Ref, White], it is demonstrated that a cylinder with a separation oriented 82 deg relative to the free stream had a coefficient of drag of 1.2. The same cylinder has a coefficient of drag of .3 when the velocity is reduced to allow the separation to occur at 120 deg [Ref. 6]. When the low pressure turbulent area is accounted for in the Bernoulli conceptualization from above it is clear that even greater drag develops in the integration.

The concepts outlined in this section are a simplification of most of the factors involved in calculating ground vehicle drag. There are no unifying theories available to date that allow the designer to analytically determine the effects of a design choice on the aerodynamic characteristics of a project vehicle. Additionally, as White admits, it has traditionally been difficult to quantify the effects of changing the geometry of one characteristic because of its downstream effects on other geometric characteristics. This leads the researcher to believe that any efforts to experimentally quantify the aerodynamic characteristics of vehicle modifications are laborious in nature, expensive to test, and provide little value to somewhat dissimilar vehicles. Throughout the rest of this thesis, the author will endeavor to show that situation is much improved when one utilizes modern CFD programs and optimization techniques.

B. CFD PROBLEM FORMULATION

The greatest benefit from computational fluid dynamics can be derived by trying to gain insight into a particular phenomena rather than trying to establish an exact value [Ref. 7]. The user is best served by attempting to establish trends in design characteristics. In developing this thesis, the author was always mindful of this approach to CFD. A baseline condition is established for a particular design and then all further designs are compared to the baseline. In this way, much of the uncertainty inherent to CFD should not confuse the researcher as the uncertainty effects all models to some degree. The real value lies in understanding and exploiting the trends that improve the models regardless of their imperfections.

A salient example, of the above mentioned modeling property, is the comparison of two different vehicles. If two separate analysts were to develop and mesh two separate CFD models, of a VW Bus and a Mazda Pickup, the results would be difficult to compare. First, the grids generated on each could be wildly different because of the two different grid building approaches. Second, the vehicle designs are so disparate that they do not allow for identification of a trend that would further the analyst's knowledge. The better use of CFD would

be to take one of the vehicles and one designer and make multiple changes to that vehicle for comparison. The designer of the VW Bus, for example, could take his mesh and carefully change the shape/curvature of the nose or the roofline. This would allow the mesh to stay relatively similar to the baseline and provide insight into the characteristics important to that vehicle. This is the approach used throughout this thesis.

CFD-ACE, produced by CFDRC, is the commercial solver used for this project to develop and analyze two dimensional and three dimensional flows. This software was chosen because it has several characteristics that were important to this project. First, it has the ability to model turbulent external flows using several different numerical turbulence models. Turbulence models are developed for particular geometries and flow conditions and may be inaccurate or unrealistic for others [Ref. 6]. Within CFD-Ace the κ-ε model was selected for the reasons stated by Versteeg [Ref. 7]:

1. Simplest turbulence model for which only initial and/or boundary conditions need to be supplied
2. Excellent performance for many industrially relevant flows
3. Well established; the most widely validated turbulence model

Other reasons for choosing CFD-ACE include: availability of training, availability and price of software, and lastly, preprocessing and post processing capabilities of the software. The most important characteristic of the CFD-ACE suite of programs is its ability to handle geometric optimizations. CFD-ACE has the advantage of developing meshes using an open-source language called Python. This language is controllable through the optimizer built into the CFD-ACE GUI. A detailed yet simple implementation of the use of CFD-ACE and its optimizer are included in chapter IV. The general methodology employed is included in the next section.

C. DEVELOPMENT OF THE SHAPE OPTIMIZATION ALIGNMENT

As mentioned in the previous section CFD-ACE was chosen partially for its ability to handle geometric optimizations. This section will give an overview of

the theory employed in the shape optimizations while chapter IV will focus on the actual implementation in CFD-ACE.

1. Software Modules that Comprise an Optimization

There are several layers of software architecture in a shape optimization using commercial CFD codes. First, a dedicated pre-processing program must be used to develop the geometry. This can be done by having the designer draw the geometry in the program or by importing an existing drawing from another pre-processor, CAD package or solid modeling package. After a drawing is imported, it is necessary to create a grid or mesh of the drawing. Utmost care must be taken to properly grid the drawing so that it is accurate, and stable at its baseline geometry as well as at the extreme ends of the design variables in optimization. Otherwise, the optimization will not converge on a successive iteration and all of the previous iterations will be for naught.

The next step is to specify the physics of the flow in the actual CFD solver as well as the solution technique. This specification has to work in concert with the grid that the designer has chosen. There are few hard and fast rules that will guaranty convergence aside from those learned in solving similar problems to the one you are trying to optimize [Ref. 7]. In this step the flow characteristics such as turbulence, velocity, boundary conditions, initial conditions, fluid type, solver relaxation, etc. are specified to match the nature of the physics that the designer is interested in.

Once the flow is solved it will be necessary to compute the objective function or cost function for the optimization. In some simple situations it is possible for the solver to do this automatically but in most cases it will require the designer to extract the cost function for use by the optimizer. This is true for both parametric studies as well as optimizations.

The last software module to set up is the optimizer. This is a simple process once all of the functions of the other software components are known. The optimizer must be able to manipulate the design variables effectively as well

as retrieve the objective function or cost function. This means that geometry and meshing module must be activated by and accept design variable inputs from the optimizer. This creates the new mesh and starts the process over again. Once the second mesh is solved for the previously specified flow parameters, the new objective function value is compared to the old. Any number of optimization algorithms could be implemented for this software module. All of which, have their weaknesses and strengths in providing reasonable updates to the design variables and choosing search directions based upon the returned objective function. At this point, the student of optimization techniques should realize that first optimization iteration is complete. The number of objective function calls or iterations necessary to reach a local minimum, from this point, is dependent upon the step size and the optimization technique employed.

To review and for clarity, a Unified Modeling Language model is provided below [Ref, 8]. For those unfamiliar with UML: the class/module title appears in the top section of the class, the class attributes appear in the next lower section, the methods or operations appear in the bottoms section, and lastly, the arrows joining the classes imply an association/action.

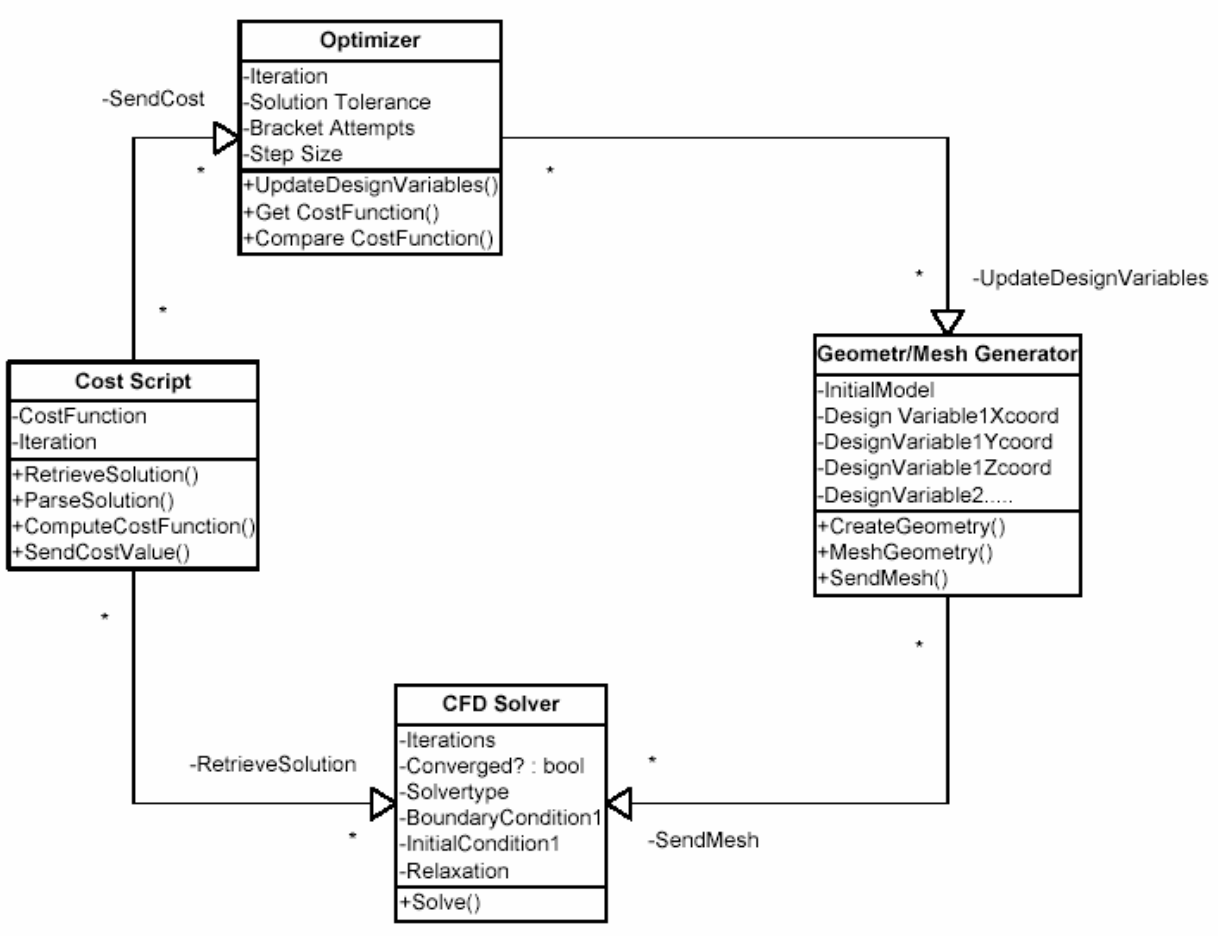

Figure 9. UML model of CFD shape optimization

2. Design Variable Strategies

There are several ways of developing an optimization strategy. The majority of this strategy is composed of selecting design variables and a technique for generating geometries and grids. All of the techniques require a compromise between solution time and detail in optimization. With a greater number of design variables it is possible to generate increasingly complicated shapes. Of course, solution time increases exponentially with the addition of design variables. Three techniques that are highly amenable to current CFD packages are the “discrete approach”, “domain element approach” and the “polynomial and spline approach” [Ref, 9].

The discrete approach was first considered by the author of this thesis before it was realized how long the objective function calls would take. The

discrete approach essentially entails shifting all of the grid points in the design space until an optimization is achieved. For every grid point there would then be three design variables. The models developed for this thesis had almost one million cells. So, it is easy to see that the number of design variables quickly becomes unmanageable to fully implement this method. As well, the software chosen for this thesis did not readily give access to actual mesh coordinates, nor were they accessible by the optimizer.

The "domain element approach" is essentially an aggregation of the discrete approach. In modern CFD pre processors one does not need to specify grid coordinates the way that is suggested in the discrete approach. Alternatively, one can build domains of cells that are grided using the preprocessor. The variables that control the shape of the domain then control a large number of cells and reduce the number of design variables necessary. Clearly, this does not provide the detail of manipulating individual cells, as in the discrete method, but may be a good compromise to eliminate unnecessary design variables. The drawing below shows how this is accomplished [Ref. 9]:

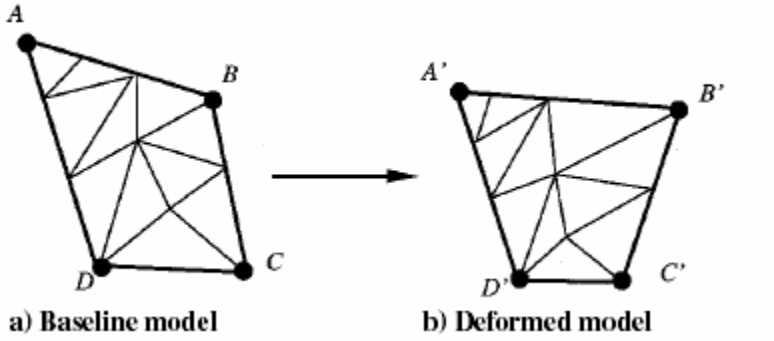

Figure 10. Domain Elements

The last applicable method is the "polynomial and spline approach" first introduced by Braibant and Fleury in 1984[Ref. 10]. Both a spline and a polynomial are easily represented by mathematical functions. They differ in practice by where the control points are specified. These control points are obviously the design variables used in this method. All of the control points are specifiable in three dimensions. As a result, the number of design variables can quickly build even when using this method. In a two dimensional design case

study on the Lockheed C-141B aircraft wing, Cosentino and Holst [Ref. 10] reduced the number of design variables from 120 to 12 by using the spline technique as opposed to a two dimensional domain technique [Ref, 10]. The spline technique can be better understood by examining the airfoil below:

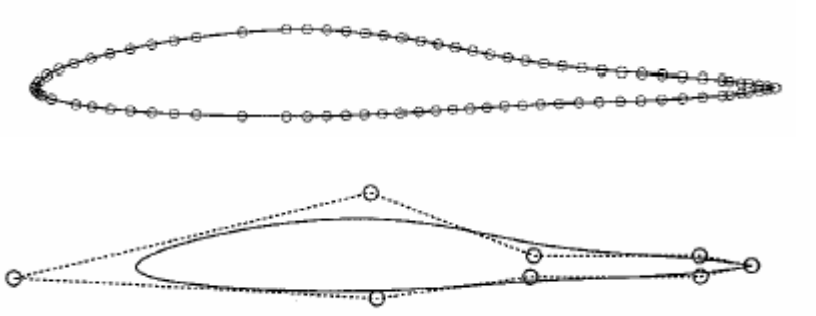

Figure 11. Spline Technique

The shapes are easily manipulated by specifying different coordinates as design variables for the control of geometry. Obviously the bottom method requires far fewer design variables and reasonably similar levels of detail.

This thesis was performed primarily in three dimensions. Computational efficiency was critical because of the three hour objective function calls. The discrete and domain element methods were considered too costly and the spline method had only been developed by previous authors in a two dimensional sense. As a result, the author chose to develop a hybrid of the spline and domain methods. This method essentially entails creating splines in three dimensions and then linking the splines to form a surface using the domain element method. This method was also chosen because the preprocessing capabilities of the available software could handle the three dimensional surface generation from splines. Elaborate shapes such as shown below can be developed using this method with relatively few control points/design variables:

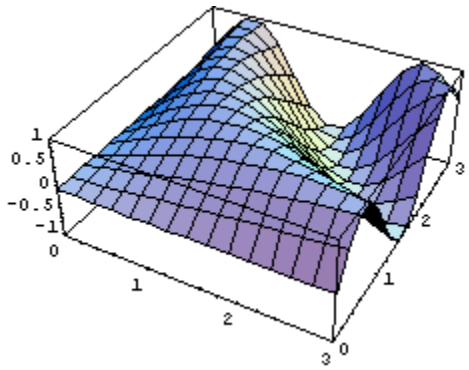

Figure 12. Hybrid Spline and Domain Element technique

Depending upon the constraints selected for the control points it should be readily evident that almost any surface of interest to automotive or aerospace engineers can be built using this method. The author was able to perform the optimization for this thesis using only four design variables.

3. Optimization Algorithm Selection

The overriding concern in shape optimization using CFD is the convergence speed of the algorithm. Unnecessary function calls can be very costly in terms of time. In the instance of this thesis, many models forced the CFD solver to work for three hours to return one objective function call. Clearly one does not want to be indiscriminate then in selecting an optimization algorithm.

Unfortunately, when using commercial CFD codes one normally has to use the optimizer programmed into the CFD user interface. Most CFD codes do not allow direct access to the solver itself so one simply cannot write their own optimization code and pass variables back and forth with the CFD suite. That said, the most important limitation on the optimization algorithm is the CFD package itself.

For this thesis there were two multi variable optimization algorithms available: Powells Method of Conjugate Directions and the Simplex method. Powell's method was chosen for two reasons. First, it is an acceptable optimization algorithm for heavily non-linear problems which Simplex is not. Second, it does not require the computation of the gradient in order perform its search. Essentially, the Powell algorithm uses successive single dimension

searches along each design variable. These search vectors are then added and formed into a new unit vector to search in the “conjugate direction”. The graphic below depicts how the Powell search is performed and how it takes advantage of the parallel subspace property [Ref. 11]:

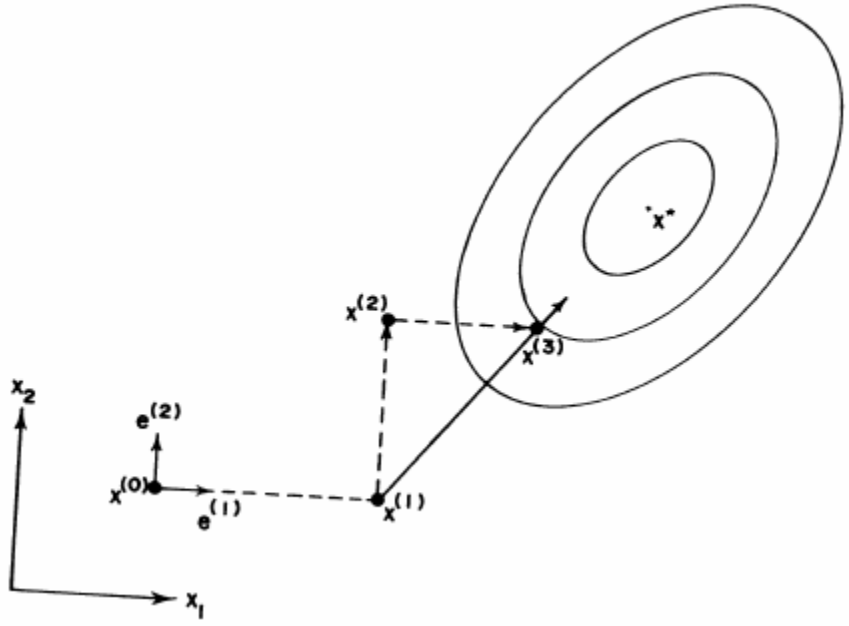

Figure 13. Powell's Conjugate Directions

Unfortunately, Powell’s method does not find the global minimum but only a local one. This is a property of most optimization algorithms. This is a mixed blessing because, optimization algorithms that are more likely to find the global minimum also require an inordinate amount of function calls, such as the genetic algorithm and simulated annealing. There is some debate over whether or not Powell’s method is the most efficient. The developers of CFD-ACE have concluded from their research that Powell’s method is the best while Belegundu and Chandrupatla [Ref. 14] claims that Rosenbrock’s method is more efficient. Rosenbrock’s method has the advantage of stopping after one failure while Powell’s appears to require two failures in the CFDRC implementation. Another unfortunate property of the CFDRC implementation is that in finishing a single variable search, minima along that direction are often calculated again, wasting a function call. This will be exposed fully in the results.

Almost all objective function calls in CFD are many orders of magnitude greater in computational intensity than the design variable generation. As such, it is extremely wasteful to run the same design variables twice. This happens frequently with Powell's method. The author would like to suggest that a step should be added to the optimization algorithm, whereby it checks all previous runs to see if the particular combination of design variables has already been attempted. This would take a fraction of a second where a wasted objective function call can often take several hours. Ultimately, Powell's method was selected because it is relatively efficient and robust and was available for use in this project.

IV. RESULTS

A. INTRODUCTION

In developing the models the author was faced with considerable difficulty in finding a suitable set of drawings, electronic or otherwise. After a rigorous search and many requests to the light truck manufacturer there were very few options. The test truck, that was to be used in the empirical analysis stage, was manufactured by Daimler-Chrysler in 1997. It is a Dodge Ram 2500 4x4 that has a very distinctive geometry. The best results would have been attained if actual electronic blueprints of the vehicle could have been used. Unfortunately, the drawings were either not made or of too sensitive for Daimler Chrysler to release to us for this project.

The best and only model found to be available was that of a very similar vehicle. A 3-D model of a second generation Dodge Ram 1500 was purchased from an online 3-D model data warehouse. The Ram 1500 shape is almost identical to the Ram 2500. Unfortunately, they vary dimensionally in width, length and height (by as much as 25%). The Ram 1500 model was carefully stretched to the actual dimensions of a Ram 2500 along all axes using CFD-GEOM. After the manipulation was completed, several key dimensions were checked manually to ensure that the model would be as close as possible. Fortunately, all of the important dimensions easily conformed to the multitude of measurements published in the owner's manual. Obviously, this method of model drawing/generation is suboptimal but provided the greatest likelihood of accuracy (especially for the three dimensional models).

B. TWO DIMENSIONAL LIGHT TRUCK SHAPE STUDIES

Two dimensional studies of light truck shapes were used first because of there relative ease in development. It was thought, by the author, that the two dimensional studies would also give some insight into the problem that could be later used to develop the three dimensional models. The two dimensional

models proved to be insightful and highly correlated with the three dimensional model in some cases. Overall though, one would not want to rely expressly upon two dimensional vehicle models as several of the models gave exceedingly erroneous results when compared to the three dimensional.

The two dimensional models were generated from the three dimensional drawing that was discussed at the beginning of this chapter. To accomplish this, all points that lay off of the models longitudinal centerline were stripped away from the drawing. It was thought that a two dimensional model would be most accurate if evaluated at the centerline as the flow should be symmetric along the center line. After the points were isolated configurations of interest were developed by connected the points. As well, a large control volume was placed around the new configuration so that flow over the configuration could be analyzed. The control volume was approximately ten times the size of the truck so that the control volume boundaries would have little or no effect on the results. Unstructured grids were then formed between the control volume and the truck. Figure 14 through Figure 49 graphically depict the effects of two dimensional configurations on local pressure and velocity.

The two dimensional models developed for this section consisted of approximately 64000 cells each. The flow over the models was specified at 30 m/s which is a very common freeway speed in the United States. As well, it is important to note that the κ-ε model was selected for the reasons mentioned in chapter 2. On average, the models converged to four orders of magnitude in approximately 45 minutes using a 2.8 ghz microprocessor and the Windows XP platform.

1. Canopies

a. Baseline Configuration

The first model of interest was a baseline model. This model will serve as a comparison for all other two dimensional models.

Figure 14. Baseline Configuration - Pressure Distribution

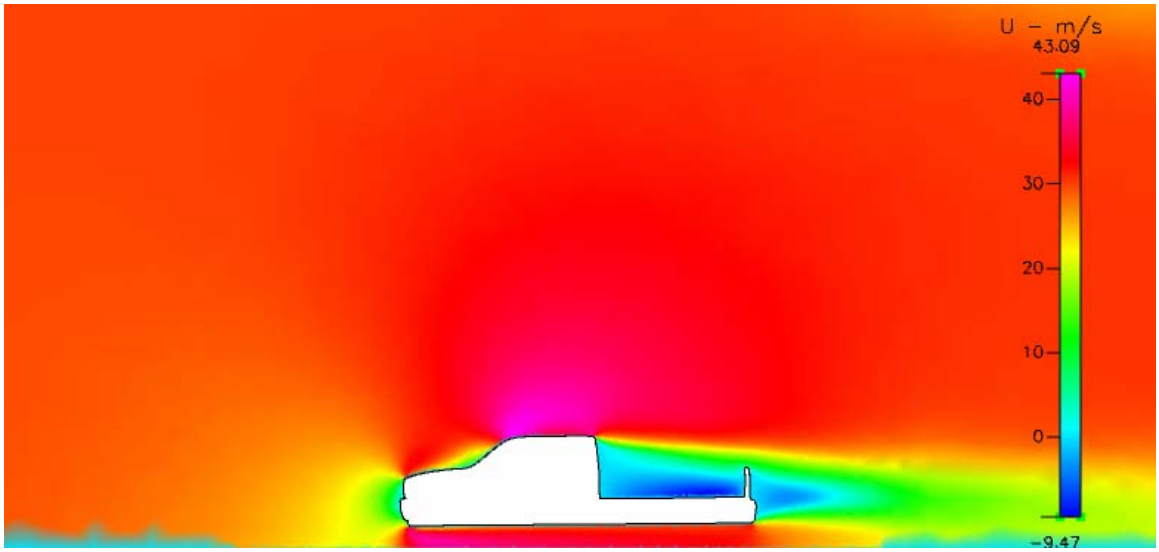

Figure 15. Baseline Configuration -X Velocity Distribution

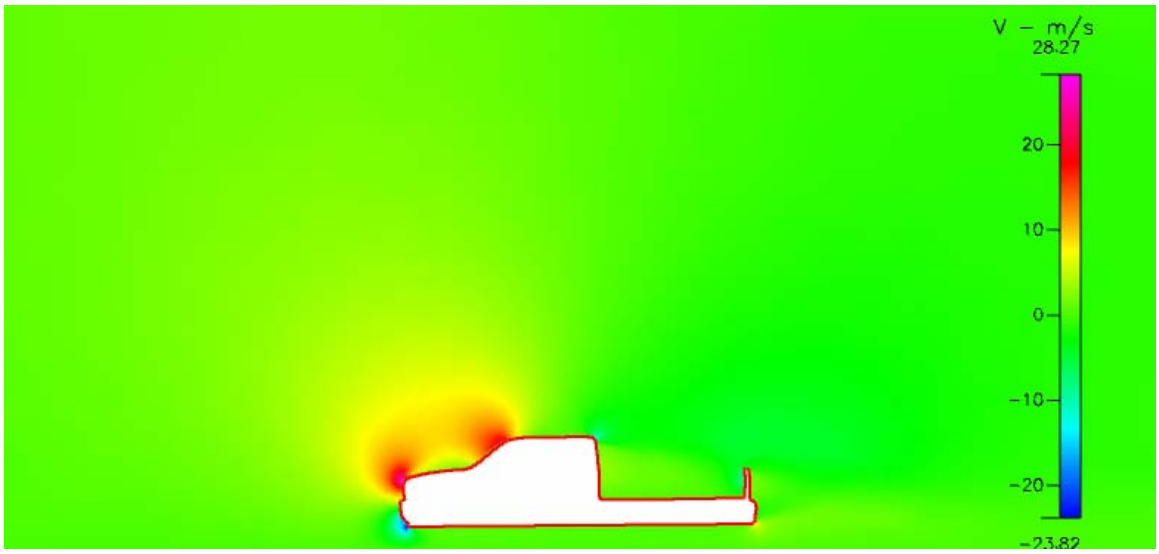

Figure 16. Baseline Configuration - Y Velocity Distribution

b. Traditional Canopy

The next configuration of interest was termed a traditional canopy. This is a highly desirable model to develop and evaluate because this type of canopy is commercially available. The roof line of the baseline was extended to the rear of the truck in a fashion similar to many commercial designs.

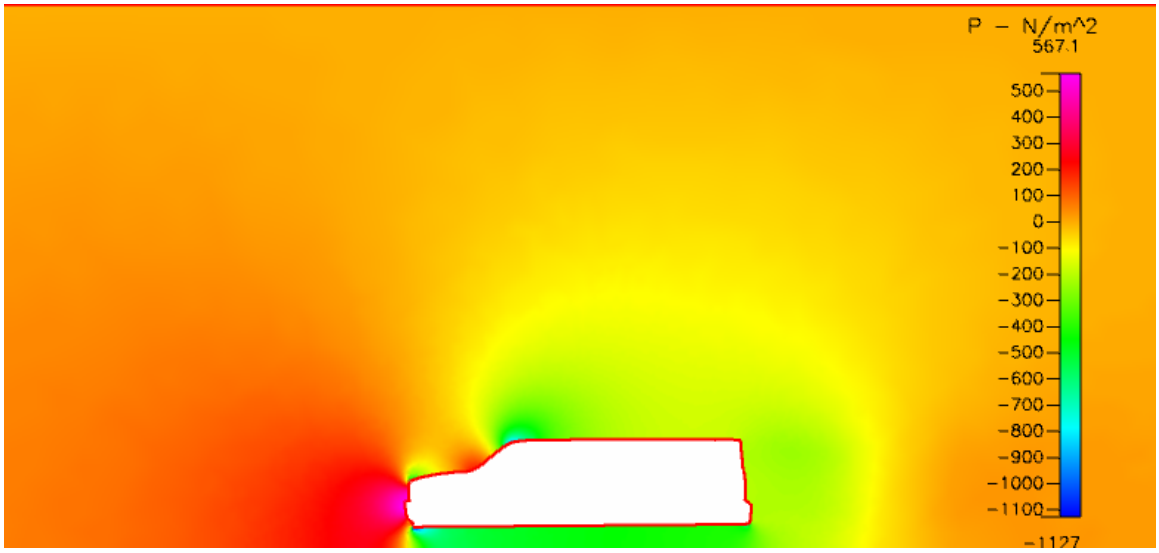

Figure 17. Traditional Canopy - Pressure Distribution

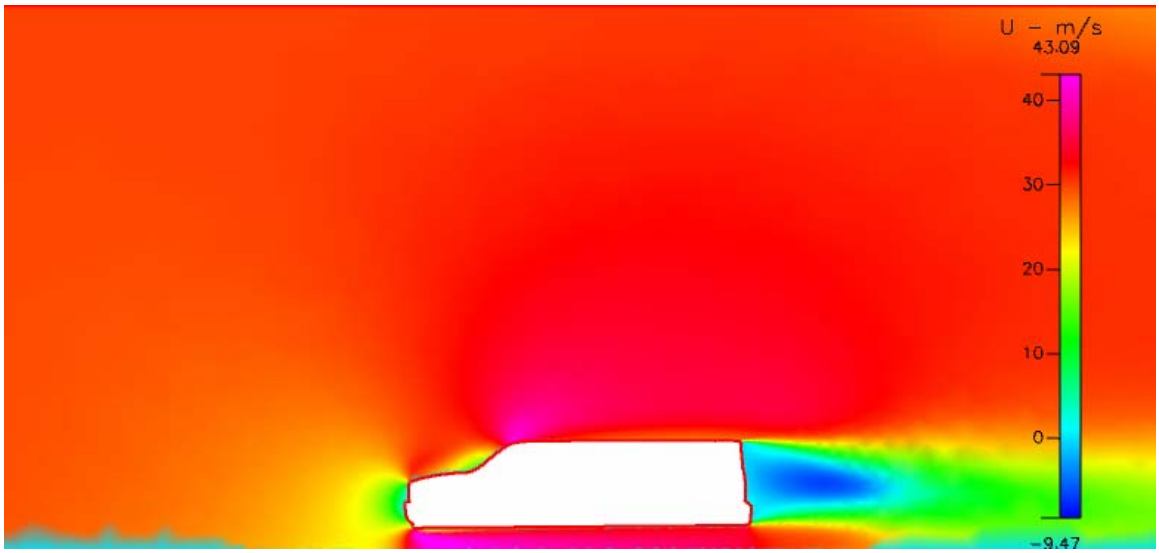

Figure 18. Traditional Canopy - X Velocity Distribution

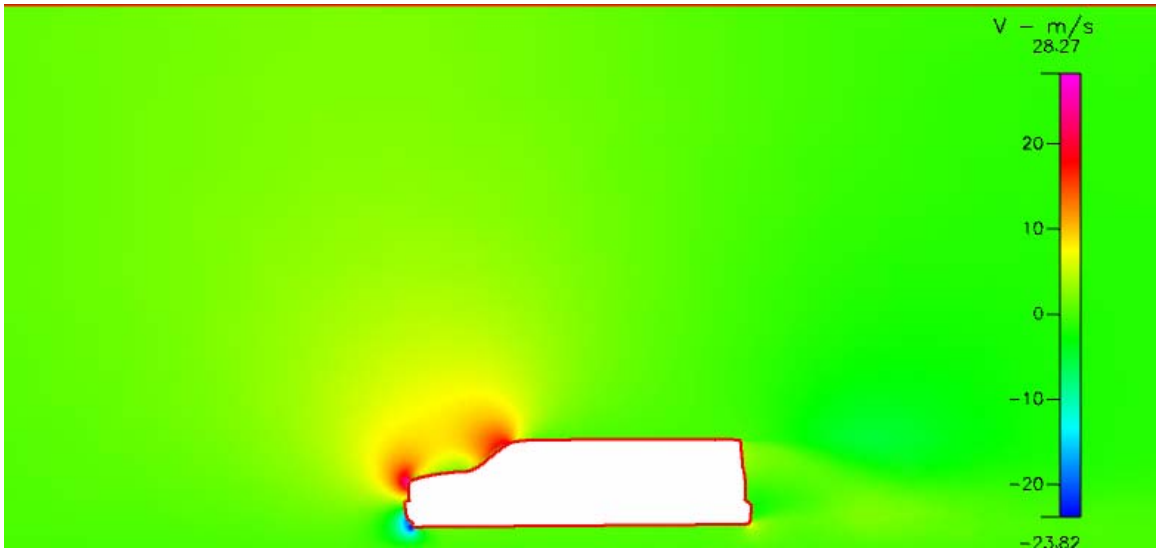

Figure 19. Traditional Canopy - Y Velocity Distribution

c. Curved Canopy

Next a canopy with a curved shape was developed as an initial estimate as to how a streamlined canopy might be shaped.

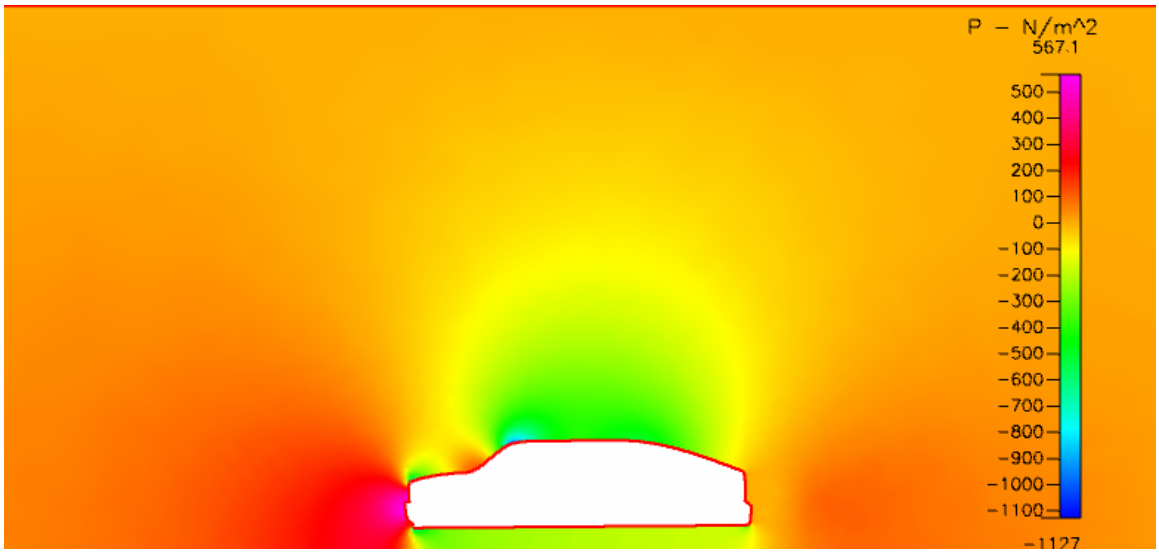

Figure 20. Curved Canopy Pressure Distribution

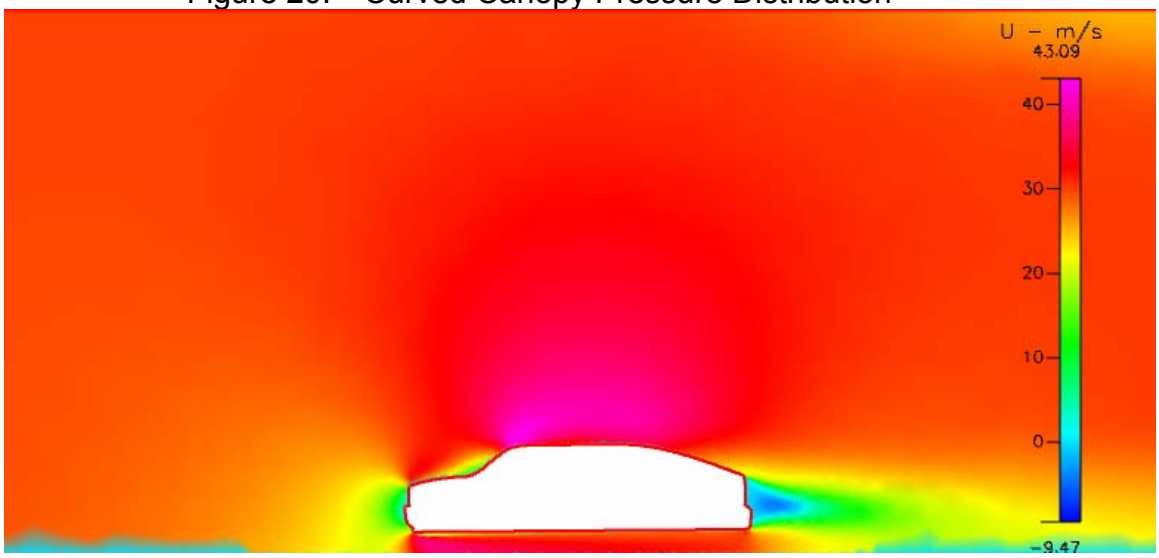

Figure 21. Curved Canopy X Velocity Distribution

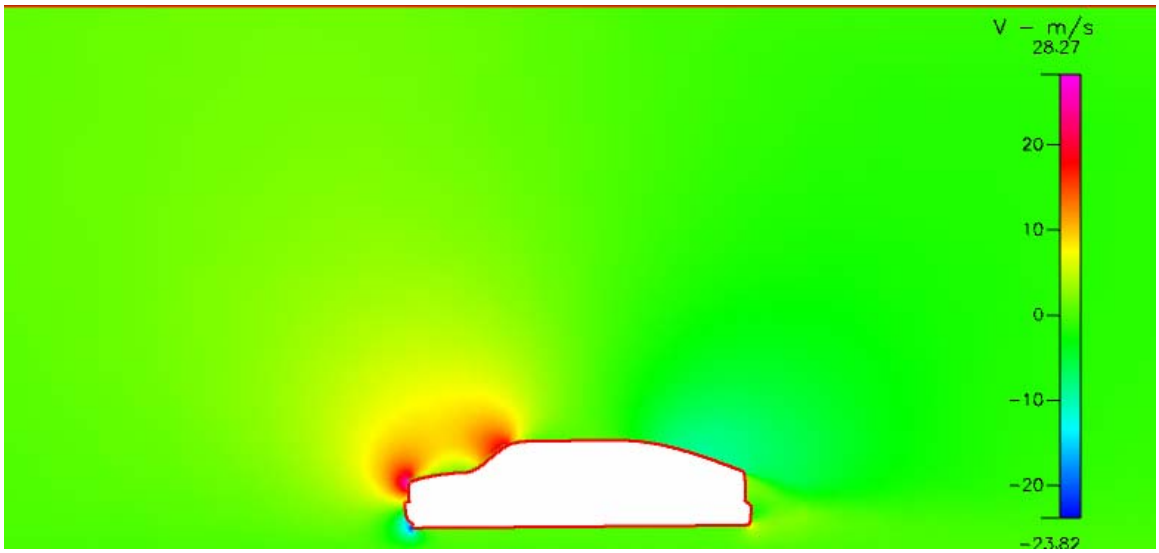

Figure 22. Curved Canopy Y Velocity Distribution

d. Tail Gate Down

A tail gate down design was developed to determine the effects of this very common practice.

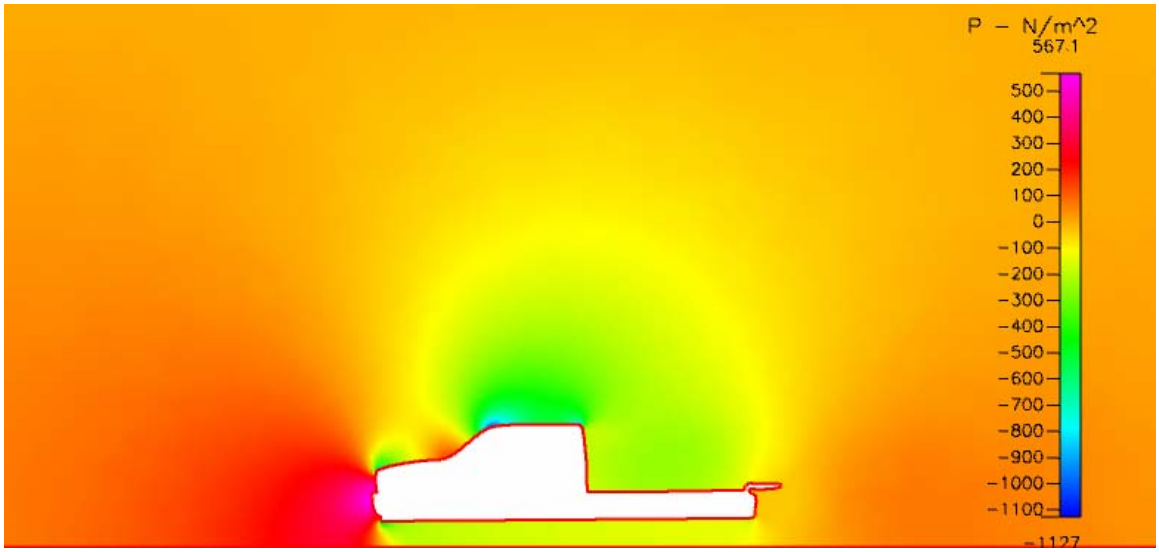

Figure 23. Tail Gate Down Pressure Distribution

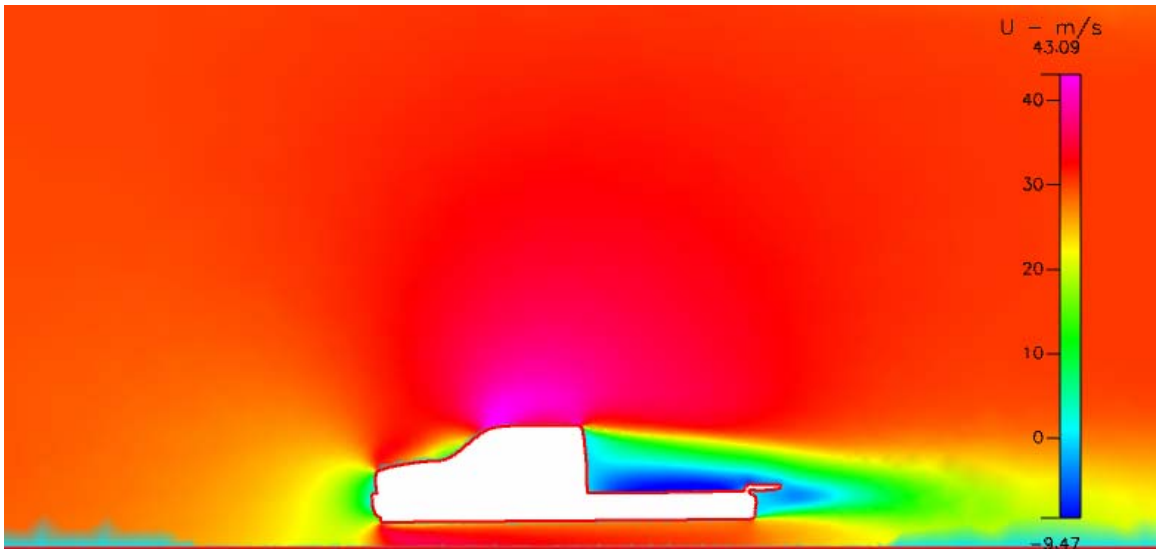

Figure 24. Tail Gate Down X Velocity Distribution

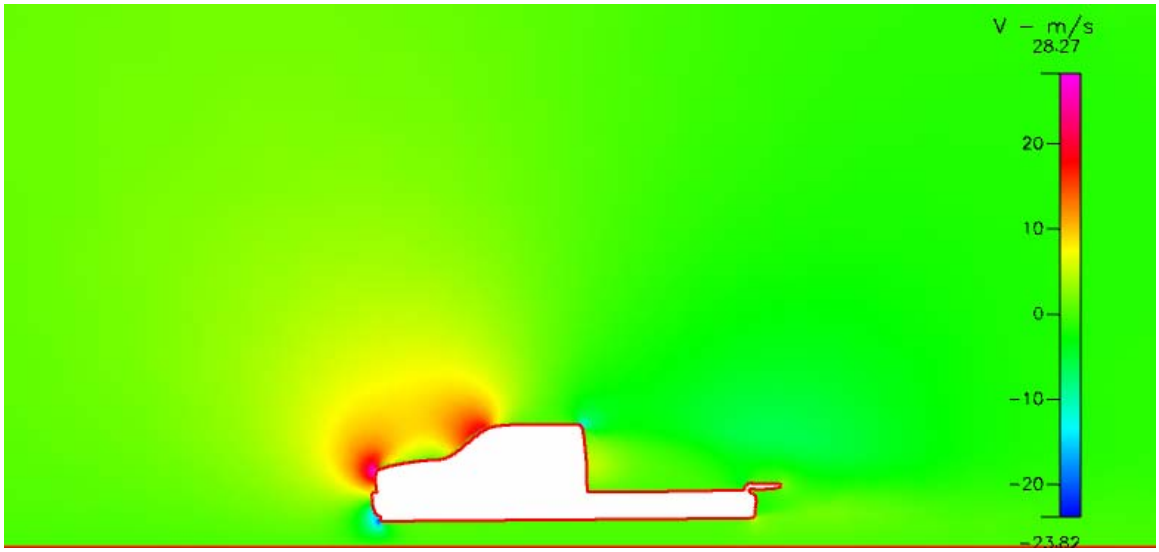

Figure 25. Tail Gate Down Y Velocity Distribution

e. Traditional Canopy with Extension

This model was developed to test the hypothesis that a traditional canopy could be improved by adding a boat tail extension. This is similar to the design employed by the NASA Dryden engineers in Figure 8 [Ref. 4].

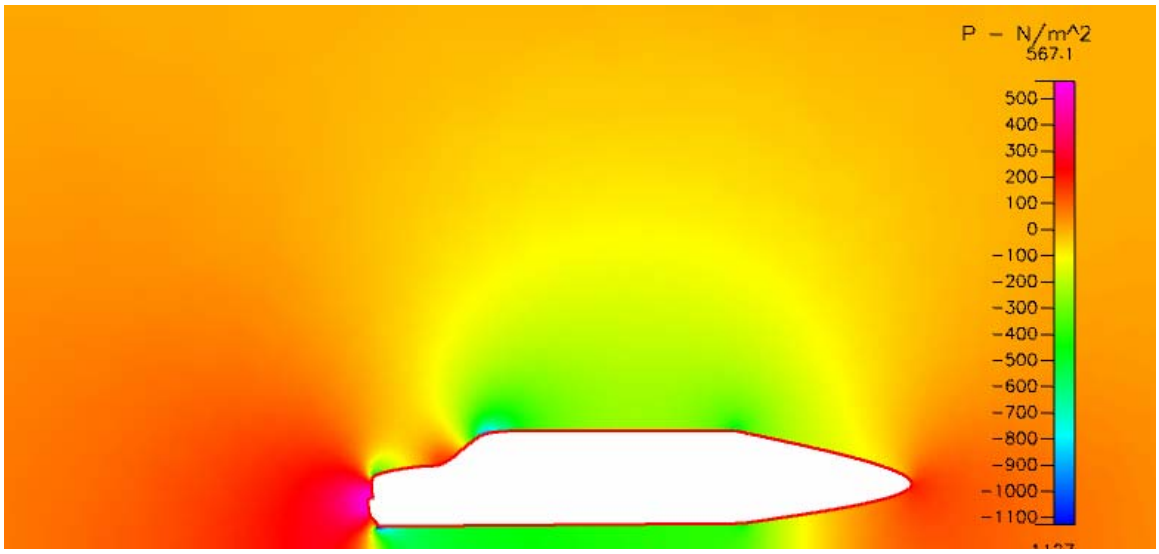

Figure 26. Traditional Canopy with Extension Pressure Distribution

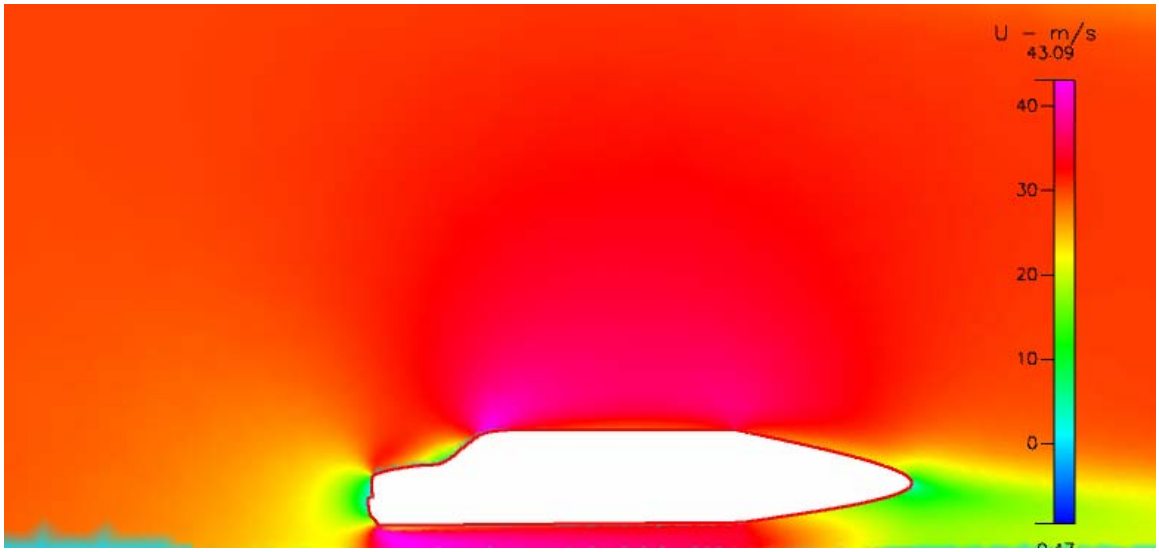

Figure 27. Traditional Canopy with Extension X Velocity Distribution

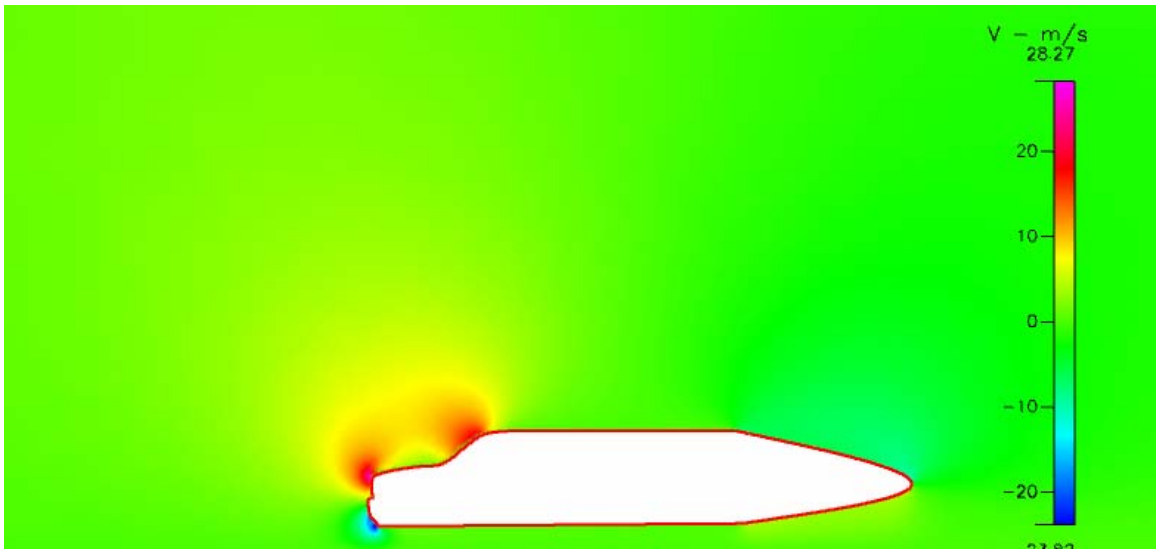

Figure 28. Traditional Canopy with Extension Y Velocity Distribution

f. Curved Canopy with Extension

Following this the train of logic from the previous three figures it was hypothesized that an extension to the curved canopy might produce better results.

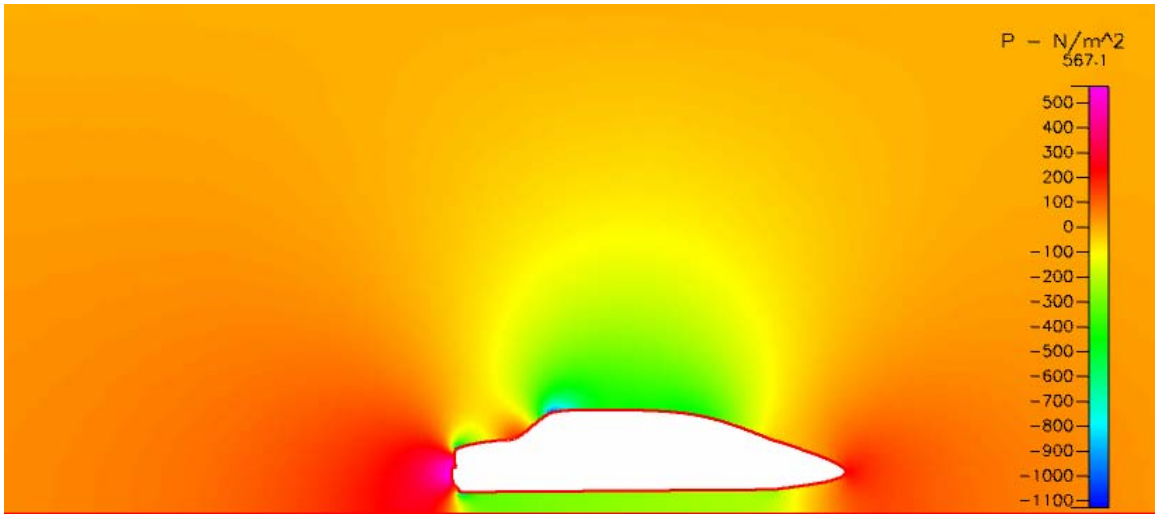

Figure 29. Curved Canopy with Extension Pressure Distribution

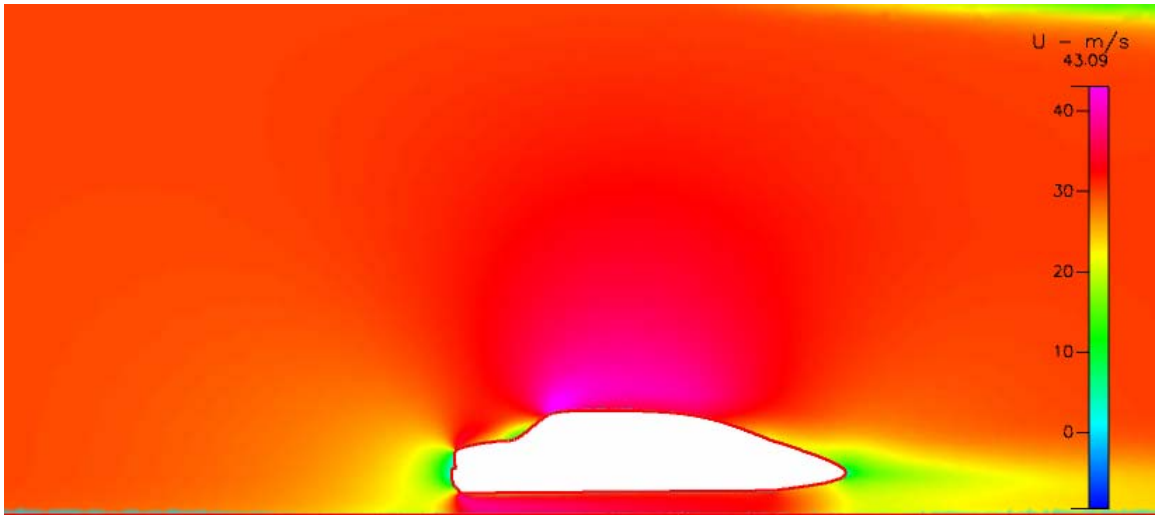

Figure 30. Curved Canopy with Extension X Velocity Distribution

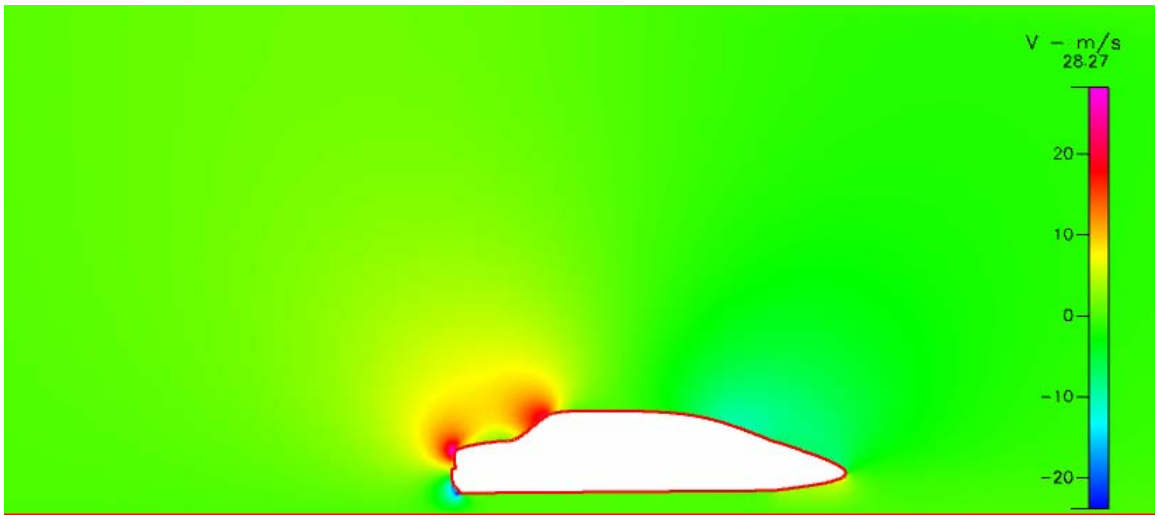

Figure 31. Curved Canopy with Extension Y Velocity Distribution

g. Baseline Configuration with Air Dam

This model was used to test the hypothesis that much of the drag is coming from the underside of the vehicle and that drag can be improved by blocking off the opening with an air dam. This proves to be the least accurate of all of the two dimensional models as the air can not slip around the vehicle in the Z direction as it can in a three dimensional model.

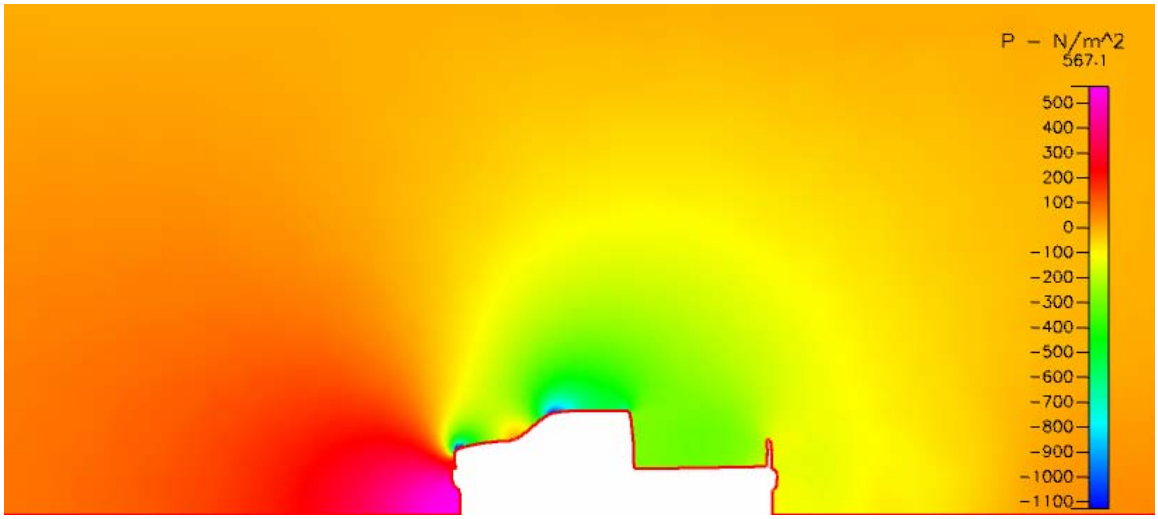

Figure 32. Baseline Configuration with Air Dam Pressure Distribution

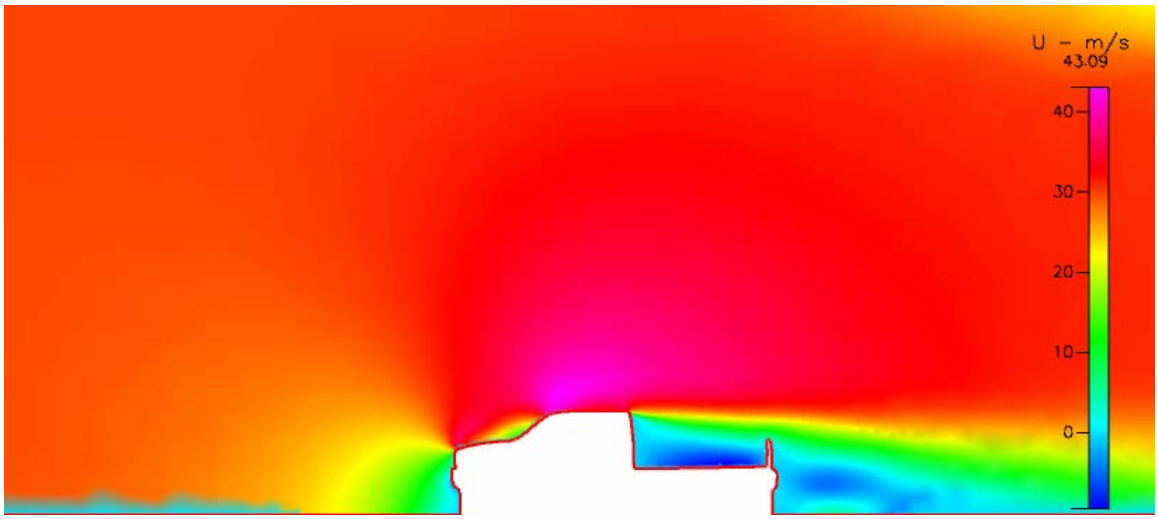

Figure 33. Baseline Configuration with Air Dam X Velocity Distribution

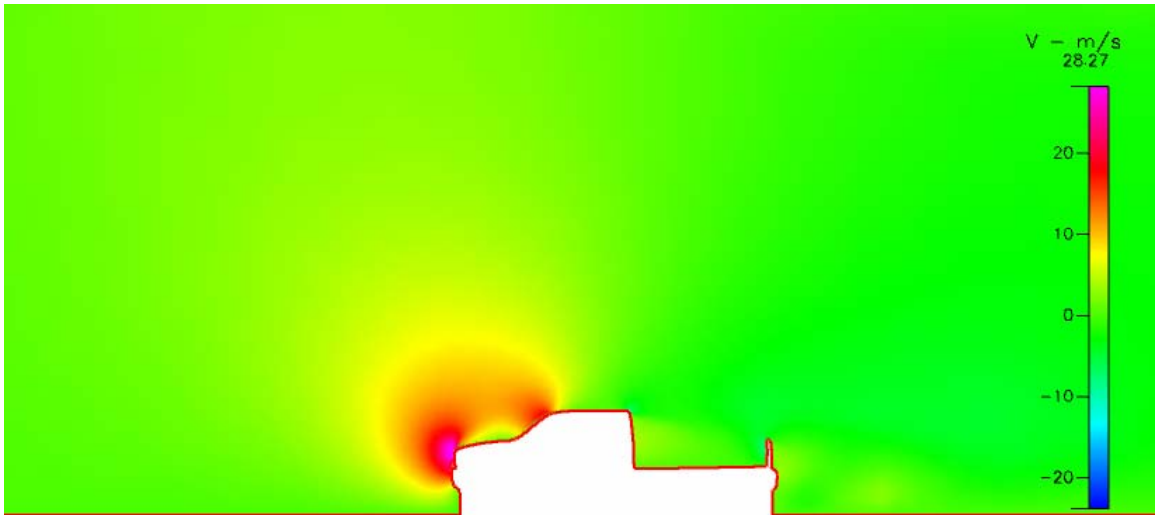

Figure 34. Baseline Configuration with Air Dam Y Velocity Distribution

h. Optimal Canopy Configuration

This design was the result of optimizing the curve that makes up the canopy with respect to drag. The method used to accomplish this optimization will be discussed in great detail in part C of this chapter.

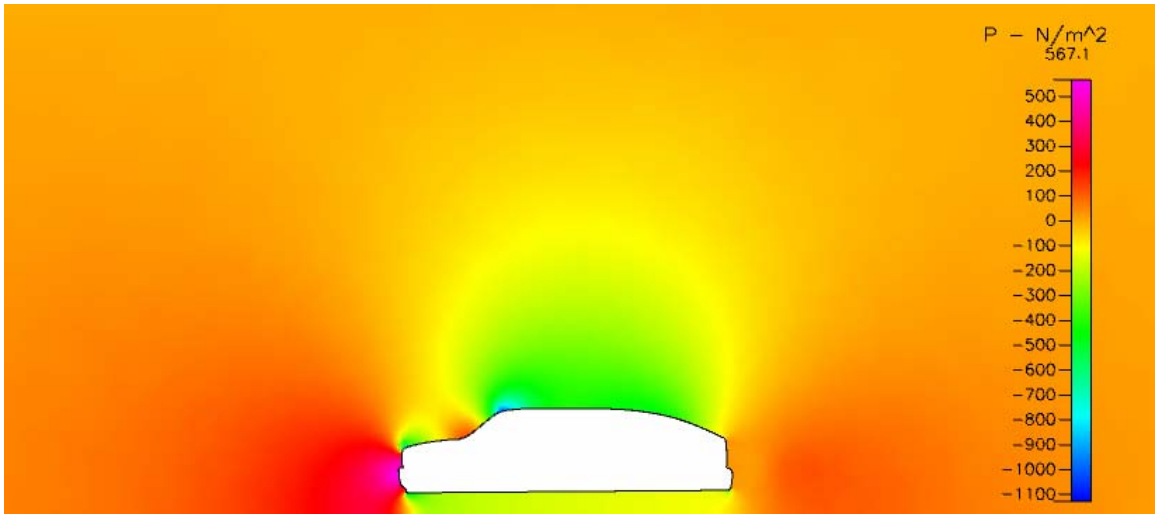

Figure 35. Optimal Canopy Configuration Pressure Distribution

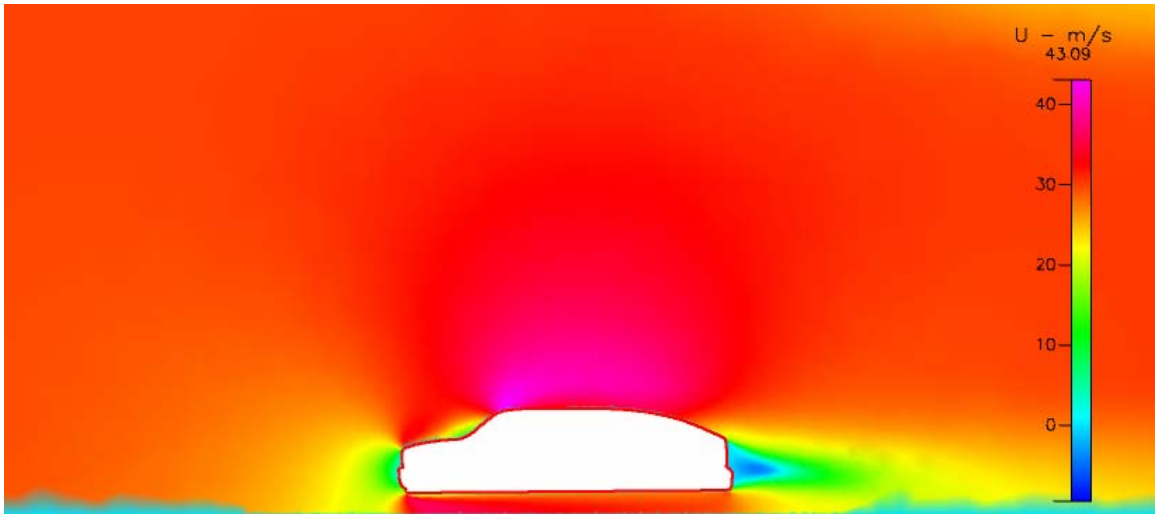

Figure 36. Optimal Canopy Configuration X Velocity Distribution

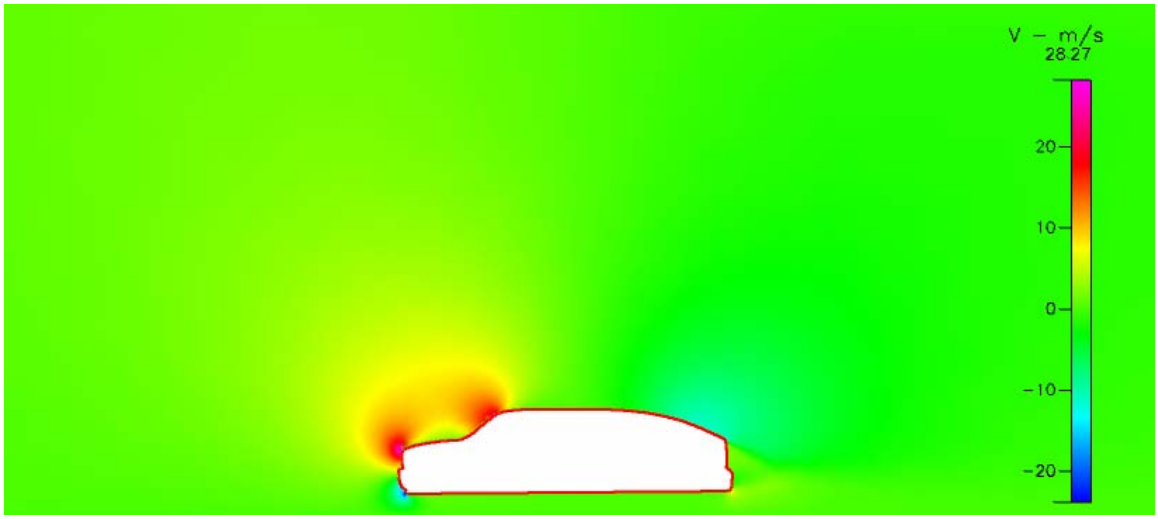

Figure 37. Optimal Canopy Configuration Y Velocity Distribution

i. Optimal Canopy with Extension

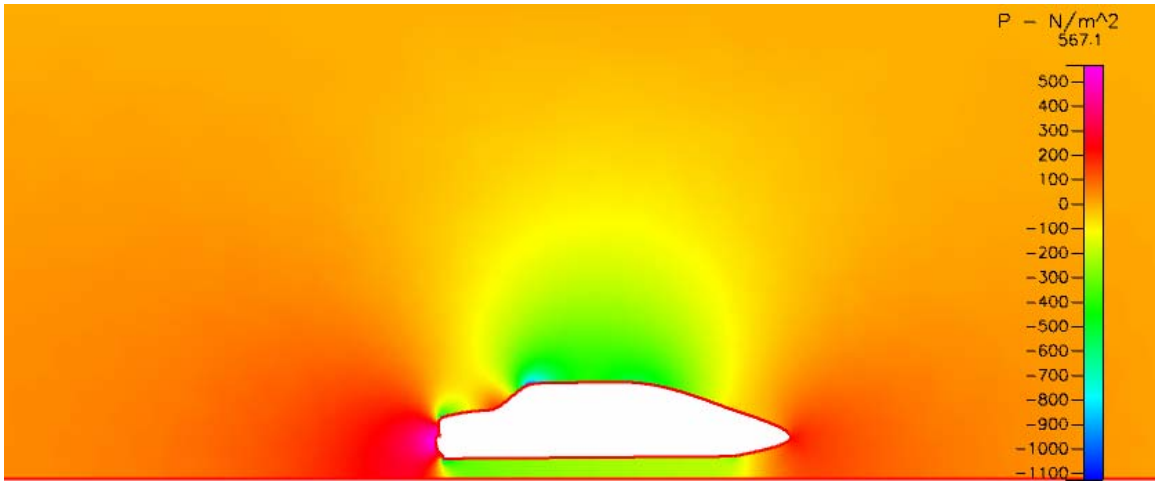

Figure 38. Optimal Canopy with Extension Pressure Distribution

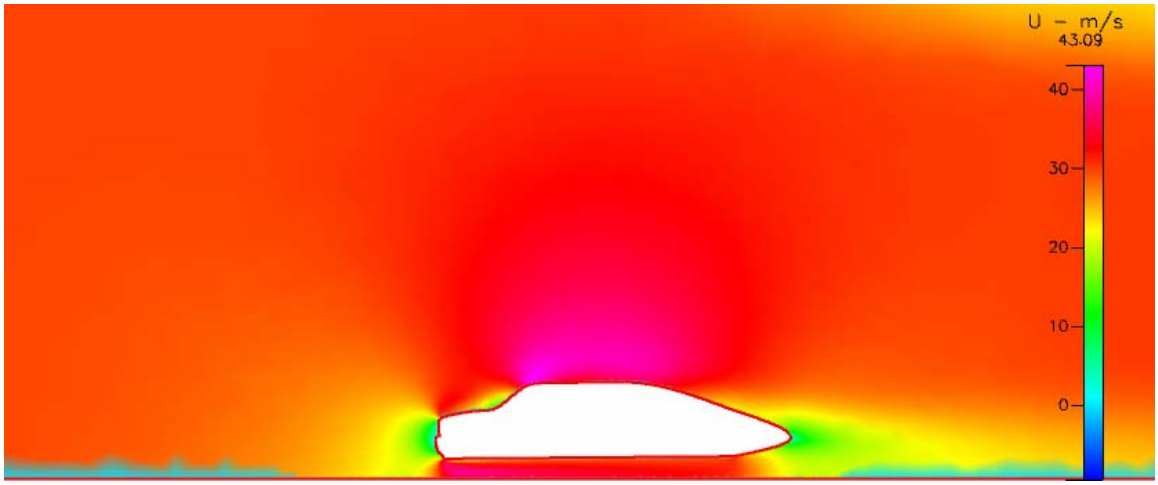

Figure 39. Optimal Canopy with Extension X Velocity Distribution

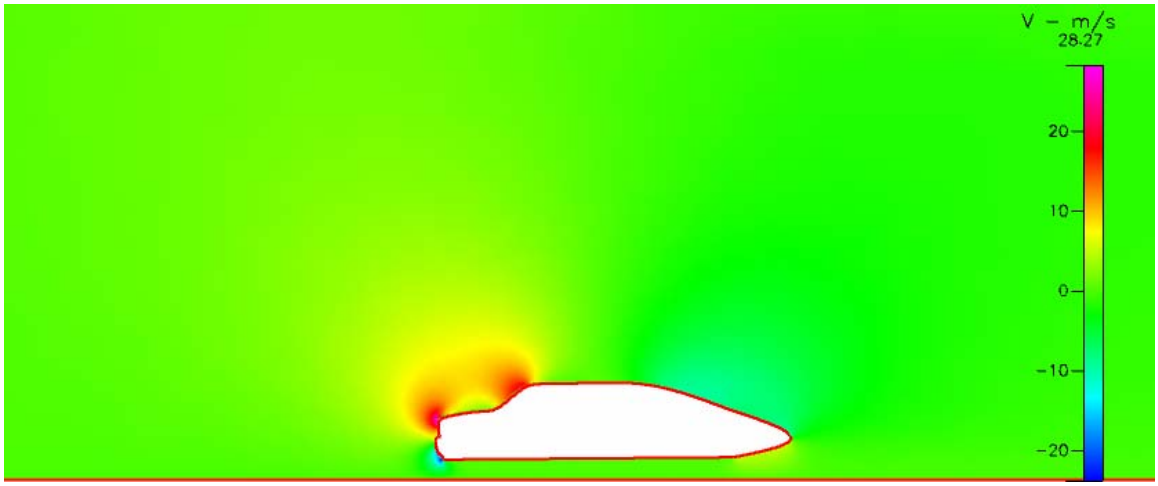

Figure 40. Optimal Canopy with Extension Y Velocity Distribution

j. Baseline Configuration with Ducting and Gate Down

The last two dimensional was one where the baseline truck had an artificial source placed at the back of the cab. The hypothesis leading to this design was based on the premise that air ducted into the bed would reduce the possibility of the effects of stagnant air in the bed increasing drag. Of course, ducting would affect the frontal area (increased drag) but it was thought that this model would provide some good initial research.

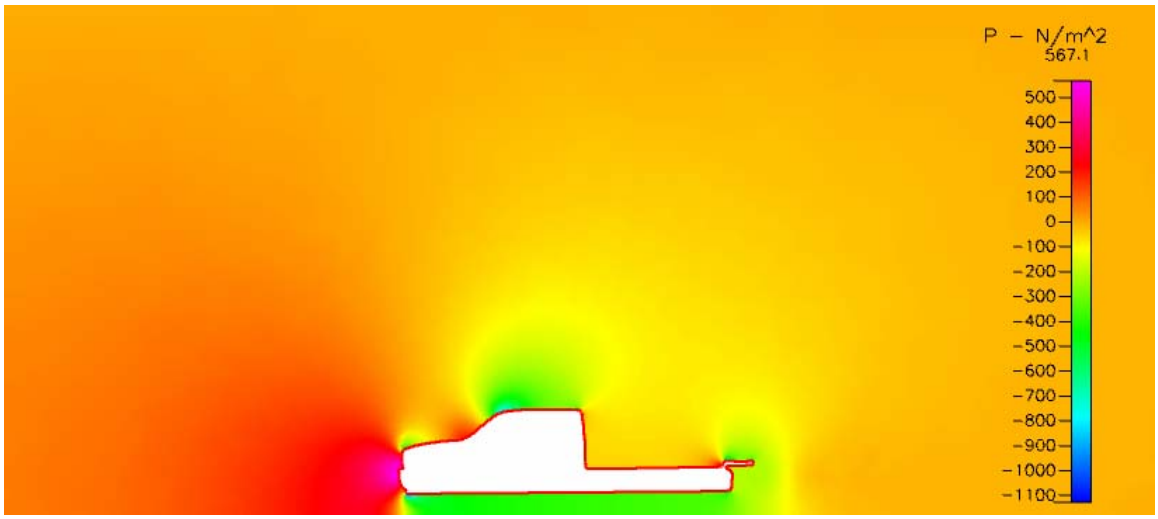

Figure 41. Baseline Configuration with Ducting and Gate Down Pressure Distribution

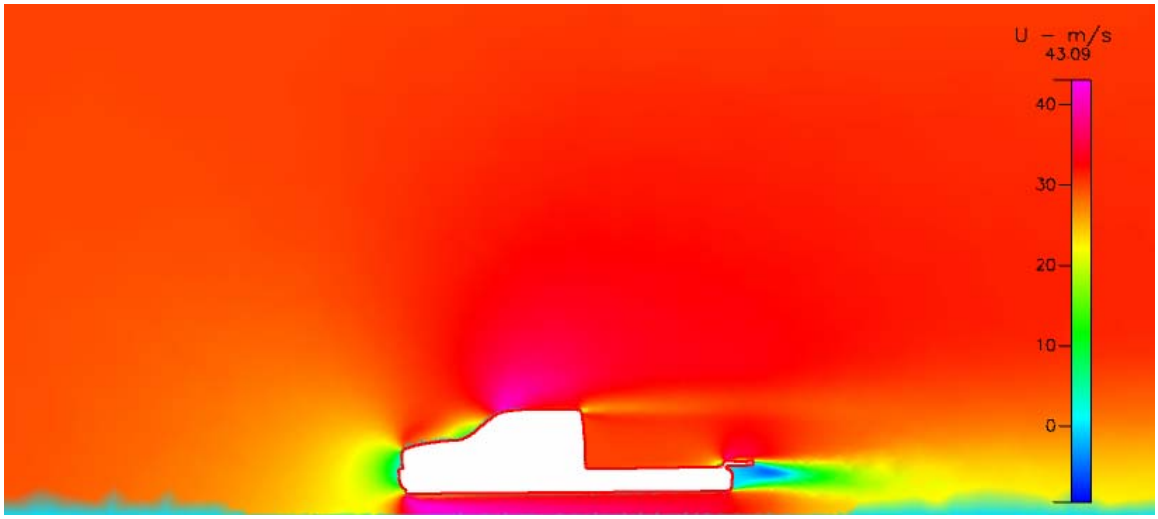

Figure 42. Baseline Configuration with Ducting and Gate Down X Velocity Distribution

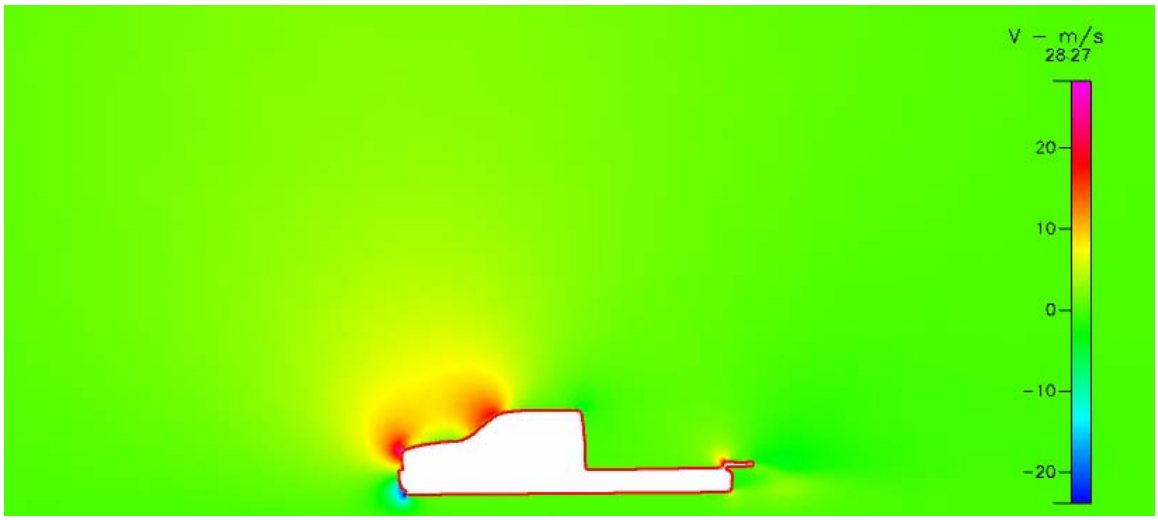

Figure 43. Baseline Configuration with Ducting and Gate Down Y Velocity Distribution

2. Underbody

In addition to the development of the air dam model (Figs 32-34), two more models were developed to analyze the effects of the irregularities in the underbody. Principally, the effects of the axles were of interest because the model was of the centerline of the vehicle. The effects of the wheels and axles are given a much more detailed evaluation in the three dimensional section of this chapter.

a. Axles Exposed

A model was developed as a baseline for a curved canopy and somewhat realistically modeled axles. It is easy to see the large effect that the axles have on the velocity and pressure distributions.

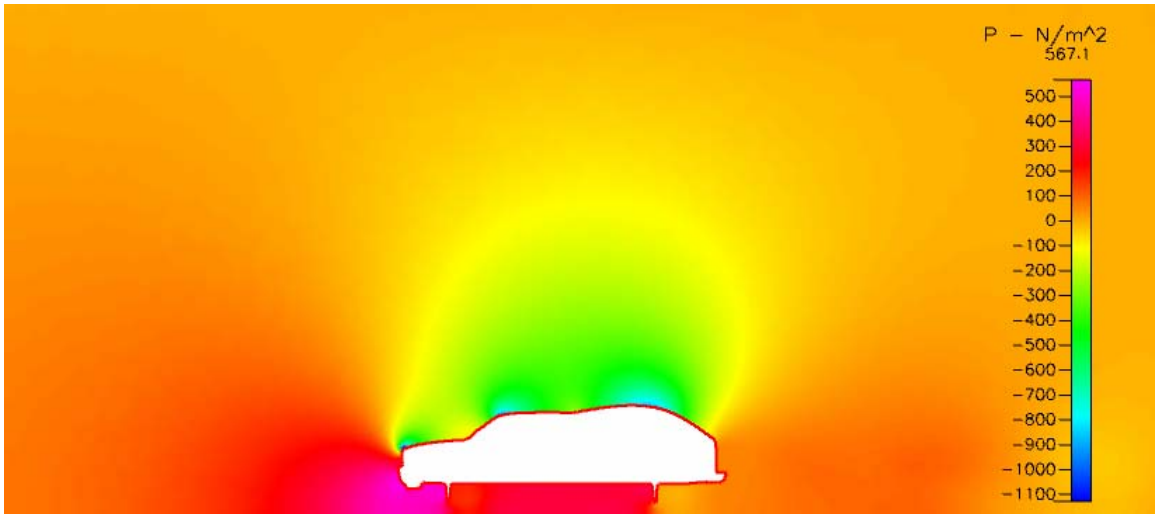

Figure 44. Curved Canopy with Axles Pressure Distribution

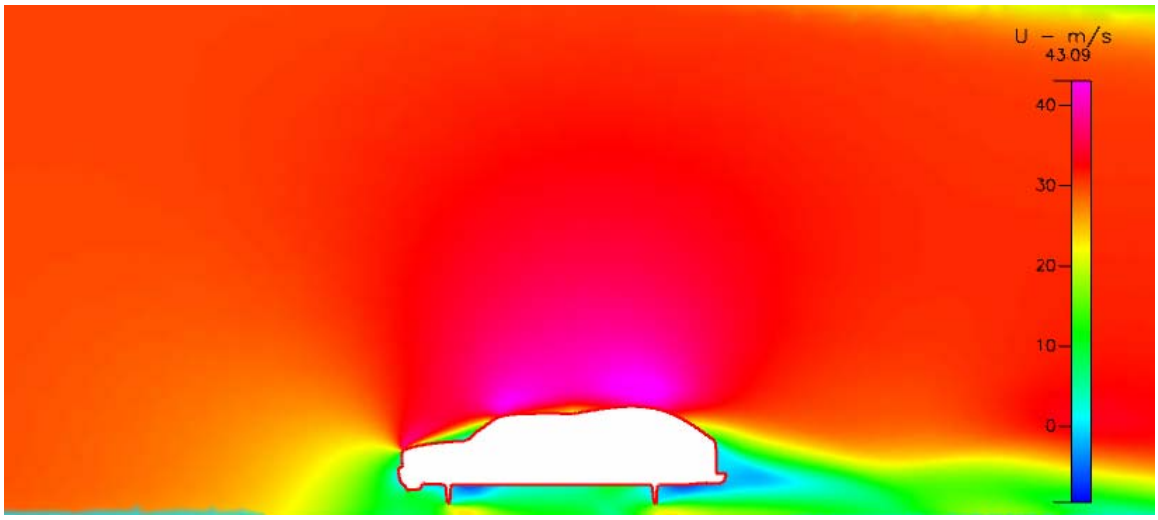

Figure 45. Curved Canopy with Axles X Velocity Distribution

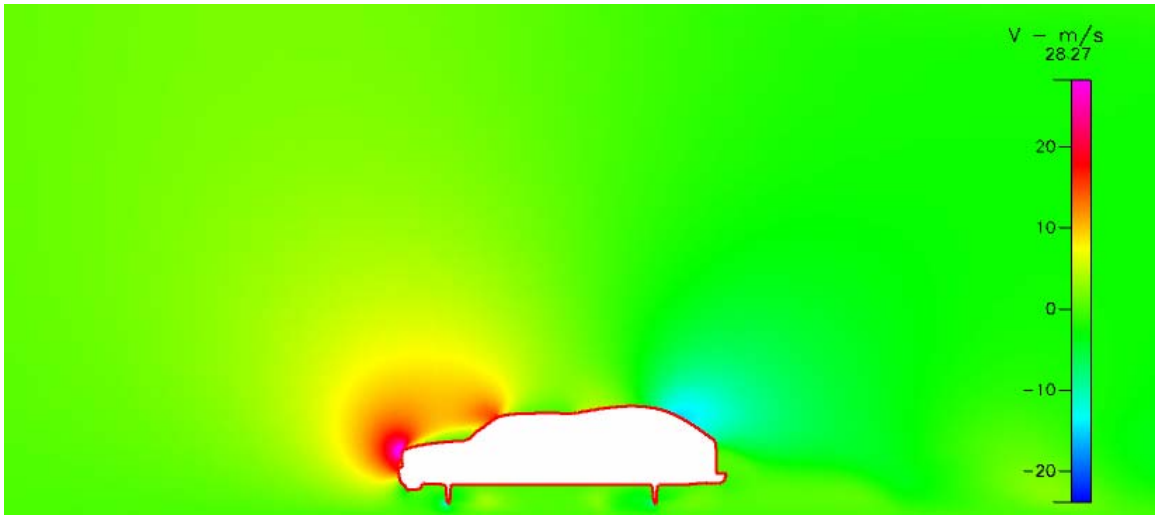

Figure 46. Curved Canopy with Axles Y velocity Distribution

b. Streamlined Axle Covers

This model was developed to see what the effect would be of streamlining the underbody so that the axles would not create such great stream disruptions.

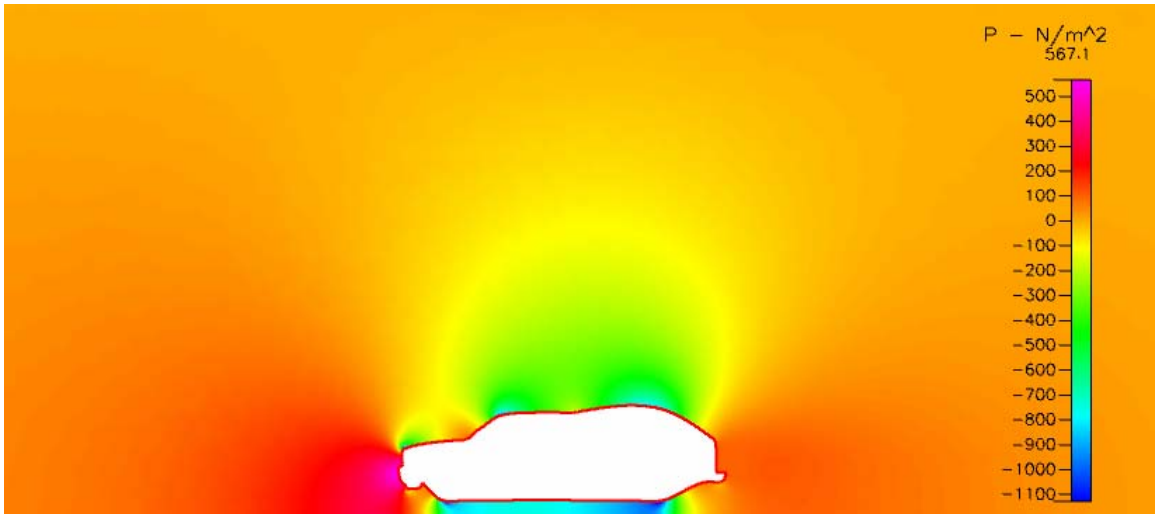

Figure 47. Covered Axles Pressure Distribution

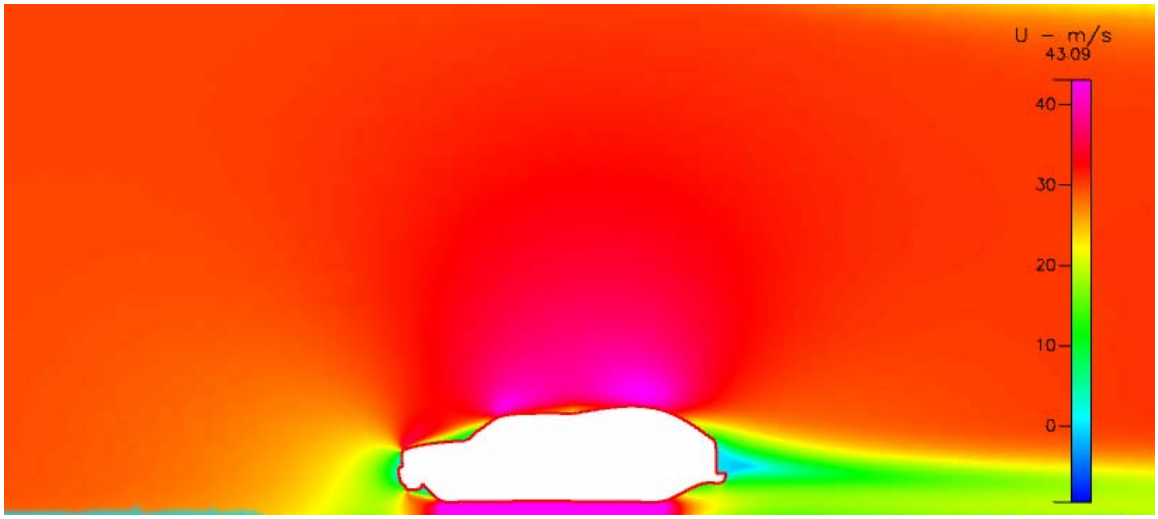

Figure 48. Covered Axles X Velocity Distribution

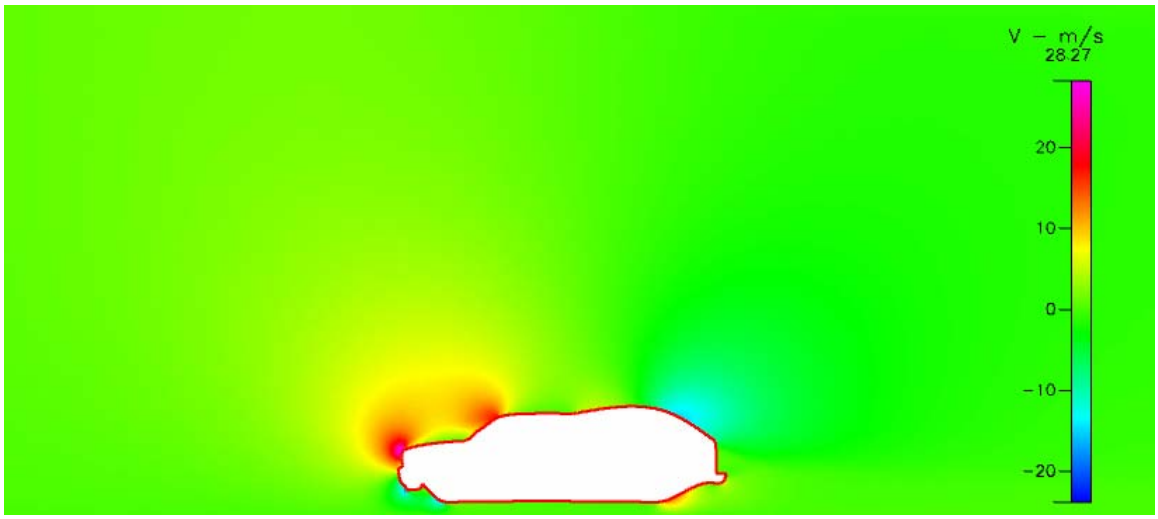

Figure 49. Covered Axles Y Velocity Distribution

3. Tabulated Comparison of Drag

The pressure and viscous forces in the X direction are tabulated below for the models shown in the previous section:

Configuration	Figures	Total force N/unit depth	% diff (from baseline)
Baseline Configuration	14-16	379.39	
Traditional Canopy	17-19	523.69	+38.03%
Curved Canopy	20-22	292.39	-22.93%
Tail Gate Down	23-25	355.28	-6.36%
Traditional Canopy with Extension	26-28	283.94	-25.16%
Curved Canopy with Extension	29-31	242.68	-36.03%
Baseline Configuration with Air Dam	32-34	492.79	29.89%
Optimal Canopy Configuration	35-37	208.30	-45.10%
Optimal Canopy with Extension	38-40	196.89	-48.10%
Baseline Configuration with Ducting and Gate Down	41-43	357.98	-5.64%
Curved Canopy with Axles Exposed	44-46	279.71	-26.28%
Curved Canopy with Covered Axles	47-49	234.32	-38.24%

Table 1. Pressure and Viscous Forces on Two Dimensional Light Truck Models

Several interesting observations can be gleaned from the two dimensional models.

1. Traditional canopies, while useful, appear to increase drag dramatically.
2. Ad-hoc selection of a curved canopy shape provides an enormous improvement when compared to the baseline and a traditional canopy.
3. It appears that dropping the tailgate provides a small but observable reduction in drag.
4. Traditional canopies are improved by adding extensions or boat tails. This observation correlates very well with the empirical research performed by NASA [Ref. 4].
5. Curved canopies with an extended boat tail are an improvement on the curved canopy alone as well as the extended traditional canopy.

6. Optimizing the canopy curve provides a further enhancement when compared to the ad-hoc selection of the curve as well as the traditional canopy.
7. Ducting does not provide a benefit.
8. The irregular shape of the axles may be one of the greatest impacts on drag as the covered axles provide a marked improvement when compared to exposed axles.
9. An air dam is very undesirable when the model is developed in two dimensions.

Several of these observations prove to be invalid when analyzed in three dimensions. The two dimensional analysis primarily provides the analyst a stepping stone toward developing three dimensional designs. The two dimensional analysis proves that the effect of the canopy and axles are not negligible when considering flow characteristics under investigation (Air, 30 m/s). In the succeeding section it will be shown that the two dimensional analysis overestimates the effects of the various geometries on what are truly three dimensional objects. This was expected by the author because the air particles are not able to flow in the z direction which would very likely diffuse the effects of any given geometry. In essence the poor designs in two dimensions won't be as inefficient and relatively good designs won't be as promising. As an obvious corollary, the optimization of a two dimensional design will not translate directly to the optimization of a three dimensional model.

C. THREE DIMENSIONAL LIGHT TRUCK SHAPE STUDIES

Once the two dimensional optimizations were well understood the author took the thesis to the next logical level: three dimensional shape studies. The three dimensional studies proved to be many times more difficult to develop than the two dimensional studies. The primary reason for this is that unstructured grids are of little use in three dimensional studies using CFDRC's software. As a result the author had to develop structured grids to define the three dimensional flows.

This was accomplished through the use of the three dimensional CAD drawing discussed in the previous sections. First, half of the CAD drawing was removed to save on computation. This is a common practice in CFD. The vehicle was to be mirrored along the longitudinal axis through symmetry boundary conditions. Once half of the CAD points had been removed a large control volume was place around the three dimensional drawing with one side of the control volume being the symmetry condition.

At this point, the development of the three dimensional structured grid becomes an art. Ultimately, the grid was refined to the point where it was grid independent yet provided manageable computation time. Grid independence is defined as the ability of a model to return the same solution regardless of whether or not the grid is made finer. The grid was coarser in areas furthest from the vehicle surface and much finer near the surface. This is commonly recognized as the best way to obtain accurate results with minimal computational time. Great care had to be taken to capture the complexity of the wheels, axles and overall shape of the body while maintaining continuity from inner volume to inner volume. This problem is even further exacerbated when one attempts to perform a geometric optimization. The analyst must understand the steps necessary to complete the entire volume of interest in order to optimize just one surface. These steps will be discussed in detail in the next section.

The three dimensional models proved to be very difficult to grid as mentioned before. They also tended to require vast computing resources. All of the models discussed in this section took over 3 hours to converge to 4 orders of magnitude. The three dimensional models consisted of approximately 600,000 cells which is roughly ten times as many cells as was required for the two dimensional unstructured grids.

Fortunately, once the first model was built and solver settings were found that would regularly result in convergence, subsequent models were much easier to produce. The author was therefore able to evaluate many different designs and combinations of designs. These included combinations of air dams, side skirts, and different types of bed covers or canopies. The graphical representation of the each design will be an isometric pressure distribution to show how each design responds in the context of the net pressure concept of drag. Additionally, the X velocity distribution, and Y velocity distribution, along the vehicle centerline, will be displayed for each of the vehicle designs considered. The Z velocity will be measured .4 M away from the centerline because the symmetry condition dictates that there not be a Z velocity along the centerline. These are shown to further the readers understanding of drag in the context of momentum changes in the stream. In general the most important graphics are the isometric pressure distribution and the X velocity distributions.

1. Baseline Three Dimensional Truck

The first design of interest is obviously the one that closely mimics the status quo. This model will be considered the baseline of comparison for all other three dimensional models throughout the analysis.

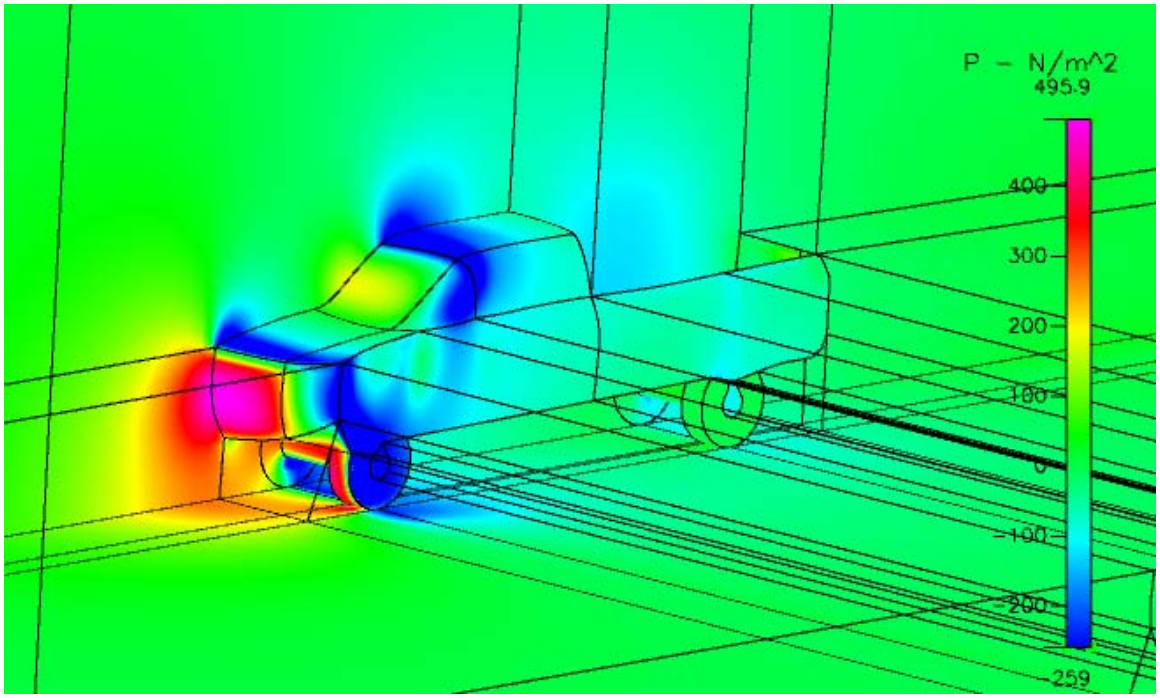

Figure 50. Baseline Truck: Frontal Isometric View of Pressure Distribution

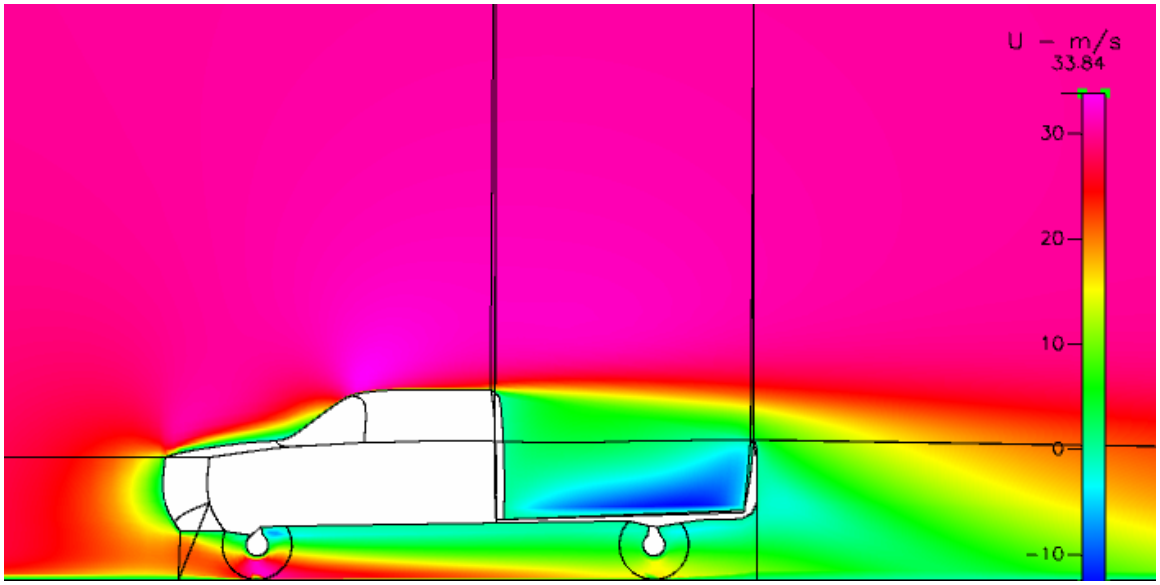

Figure 51. Baseline Truck: Centerline View of X Velocity Distribution

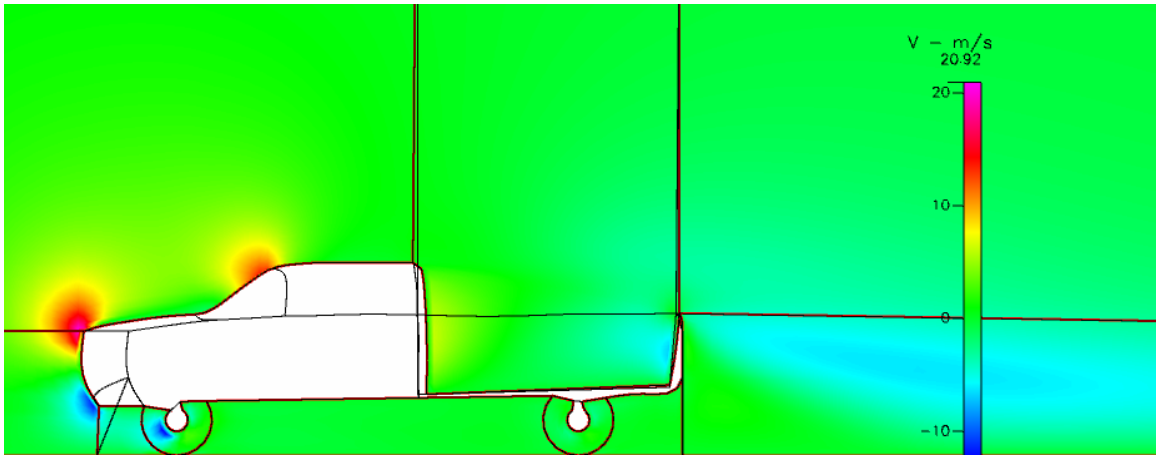

Figure 52. Baseline Truck: Centerline View of Y Velocity Distribution

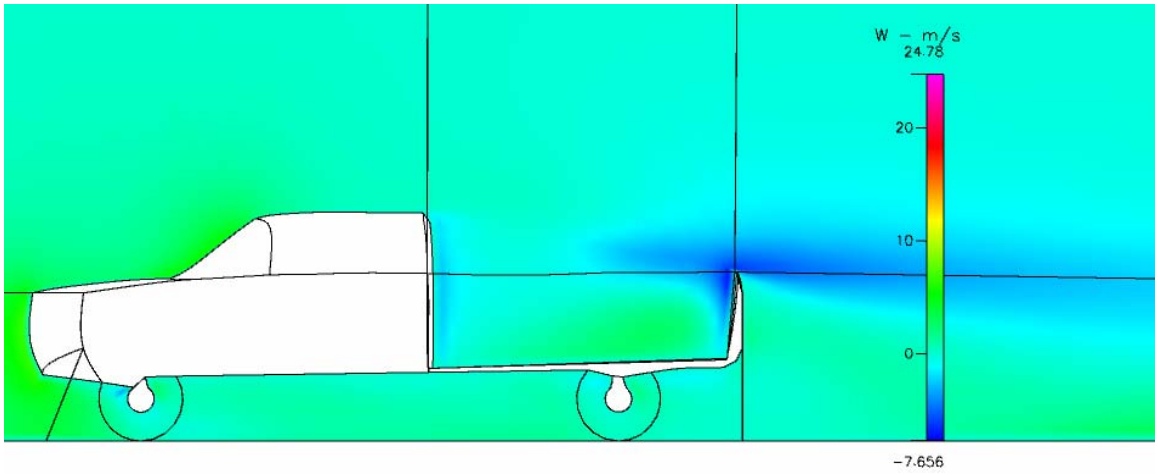

Figure 53. Baseline Truck: Z Velocity Distribution .4 M from the Centerline

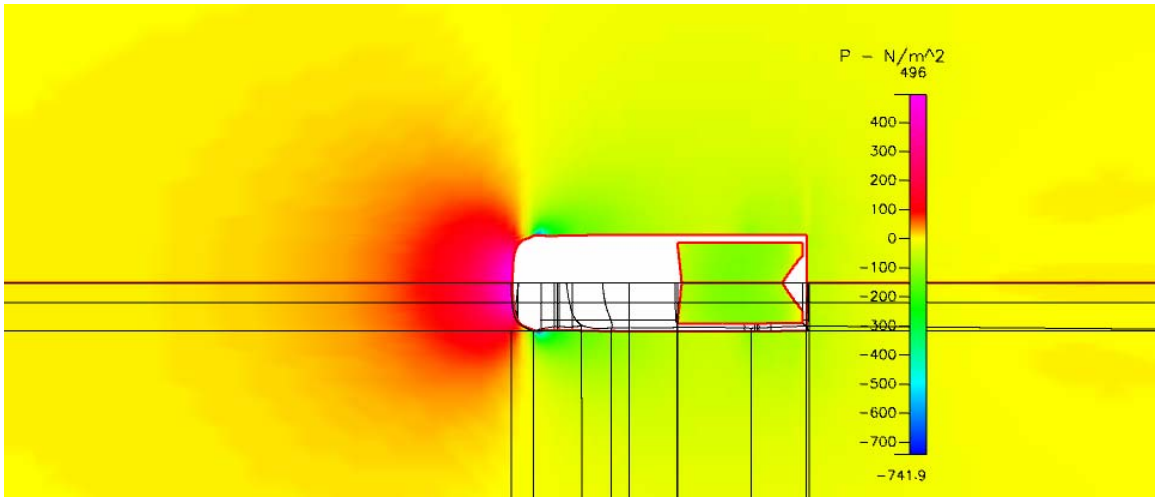

Figure 54. Baseline Truck: Pressure Distribution Top View

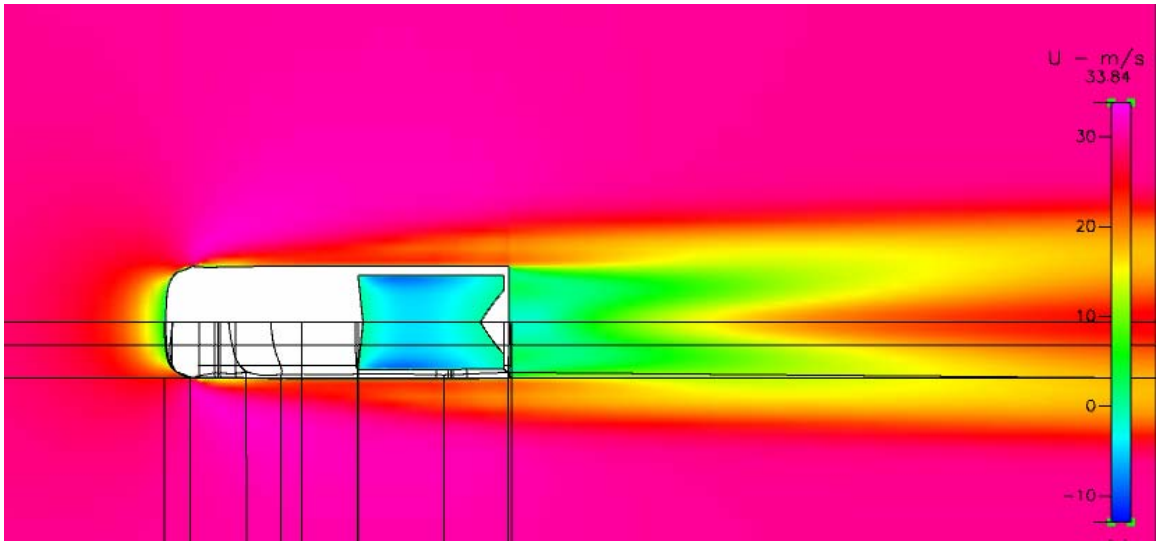

Figure 55. Baseline Truck: Top View of X Velocity

2. Truck with Air Dam

This model was developed based upon the intuition gained from observing the air dam on race cars. The goal was to see if an air dam would improve the drag characteristics. Also, this model was developed to see if the massive increase in drag predicted by the two dimensional air dam model would prove accurate.

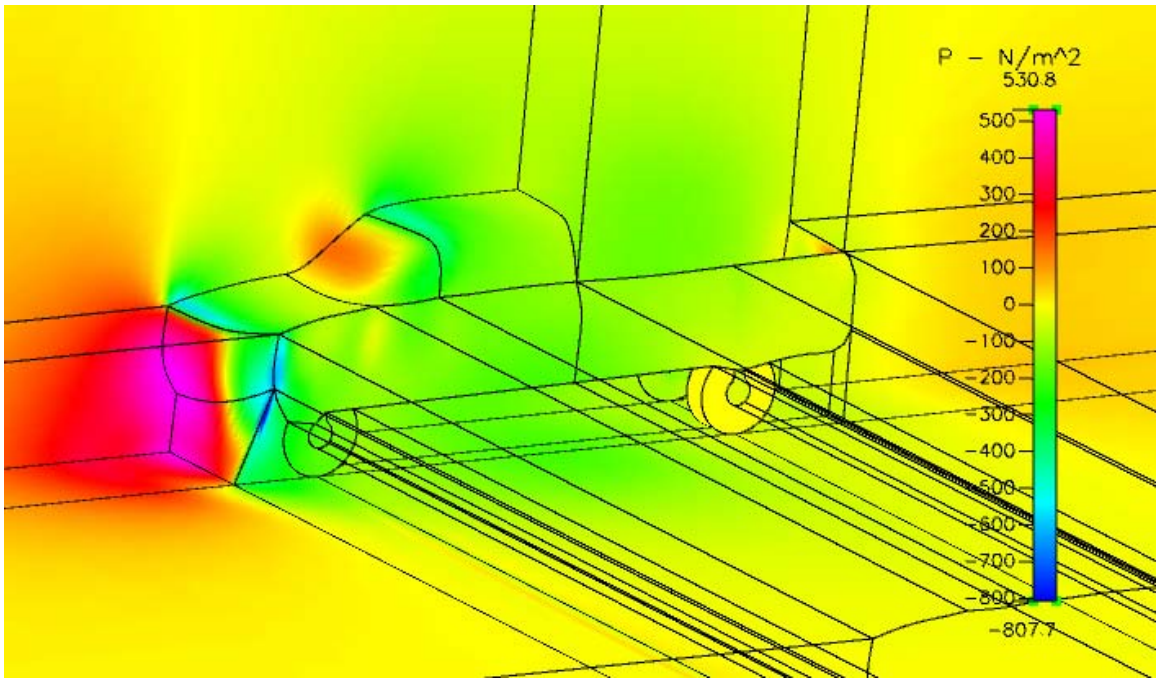

Figure 56. Truck with Air Dam - Isometric Pressure Distribution

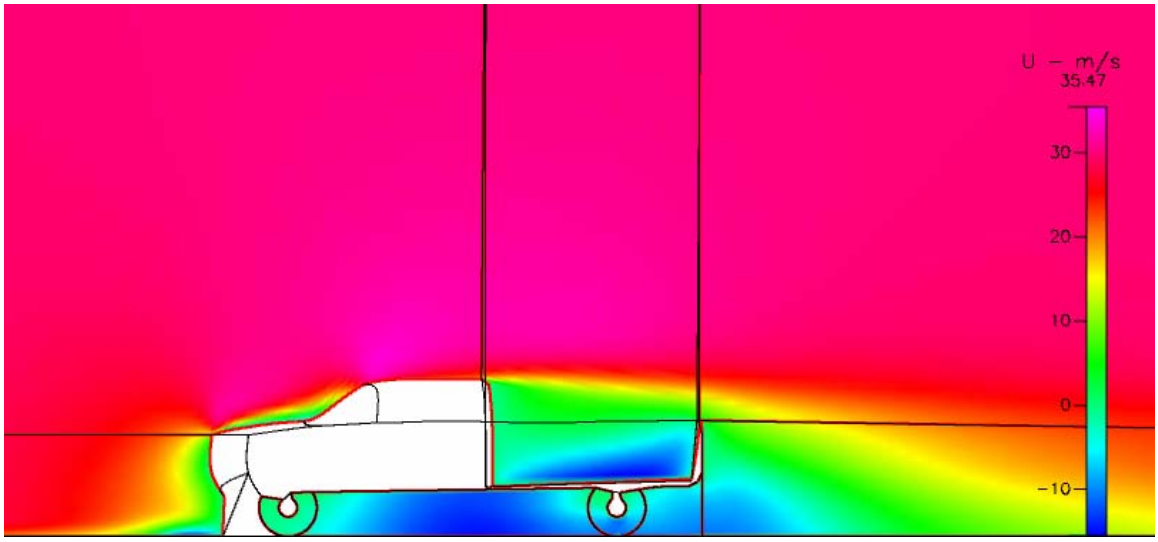

Figure 57. Truck with Air Dam: X Velocity Distribution Along Centerline

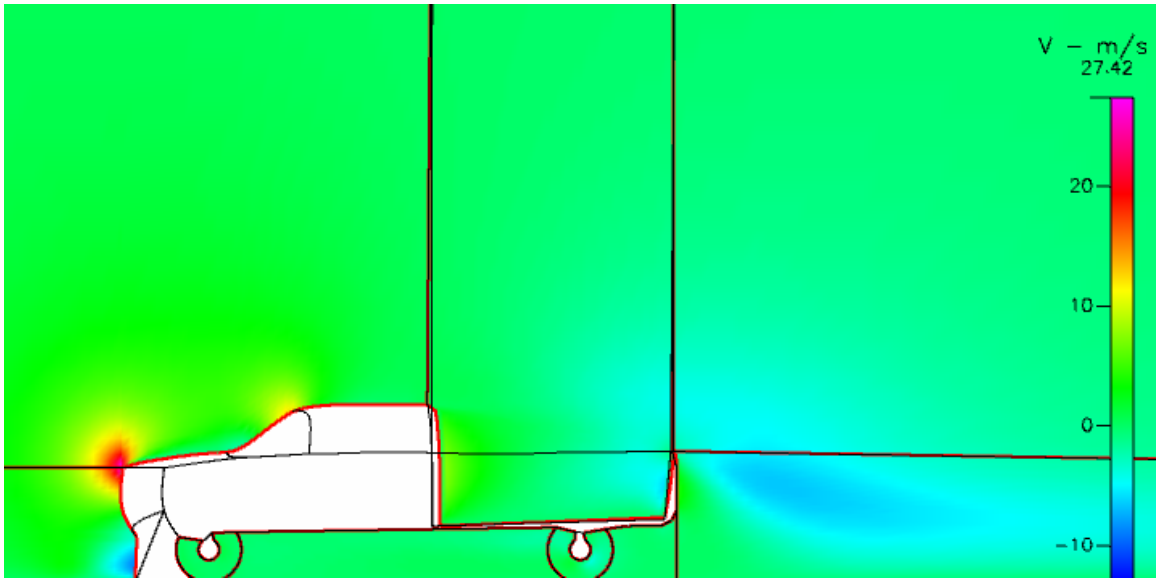

Figure 58. Truck with Air Dam: Y Velocity Distribution Along Centerline

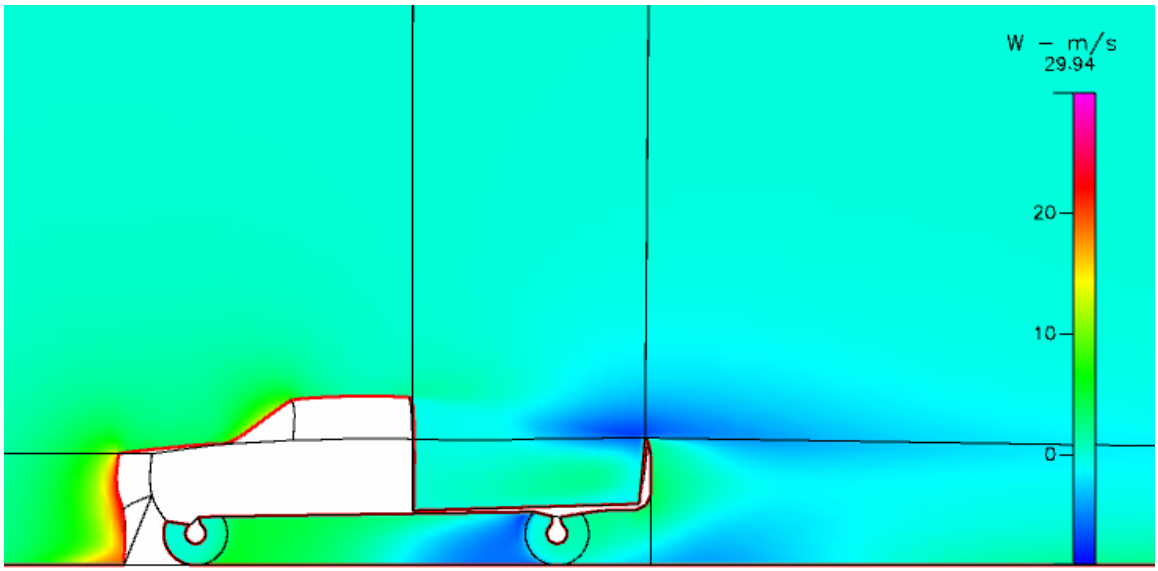

Figure 59. Truck with Air Dam: Z Velocity Distribution .4 M from Centerline

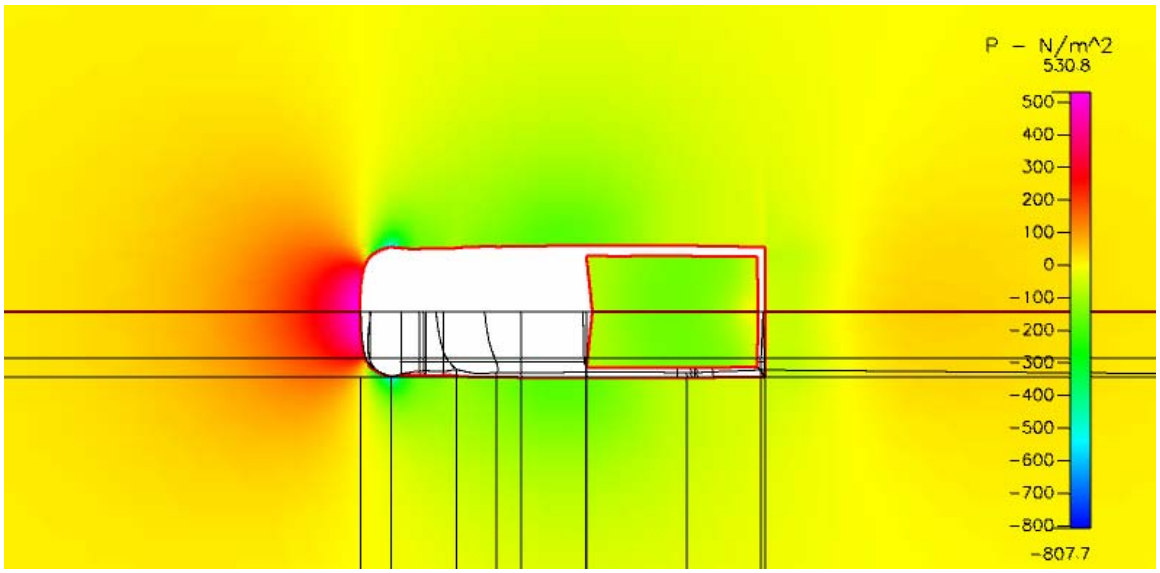

Figure 60. Truck with Air Dam: Top View of Pressure Distribution

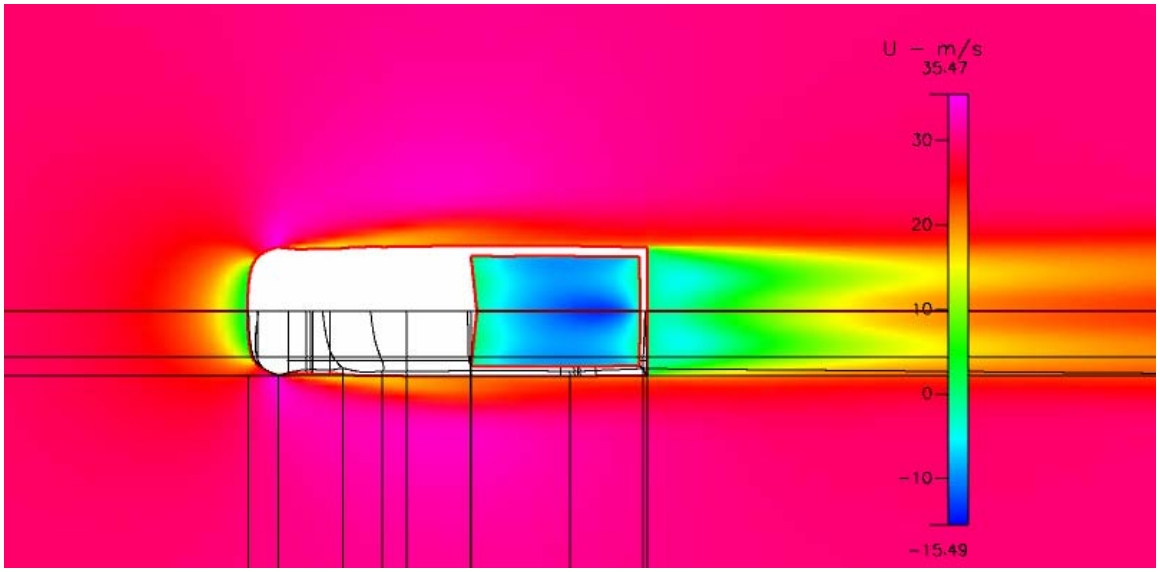

Figure 61. Truck with Air Dam: Top view of X Velocity Distribution

3. Truck with Air Dam and Side Skirt

This model was designed to see if blocking the flow after the air dam, at the edge of the wheels, would provide any additional benefit. Air is blocked completely from flowing under the vehicle in this model.

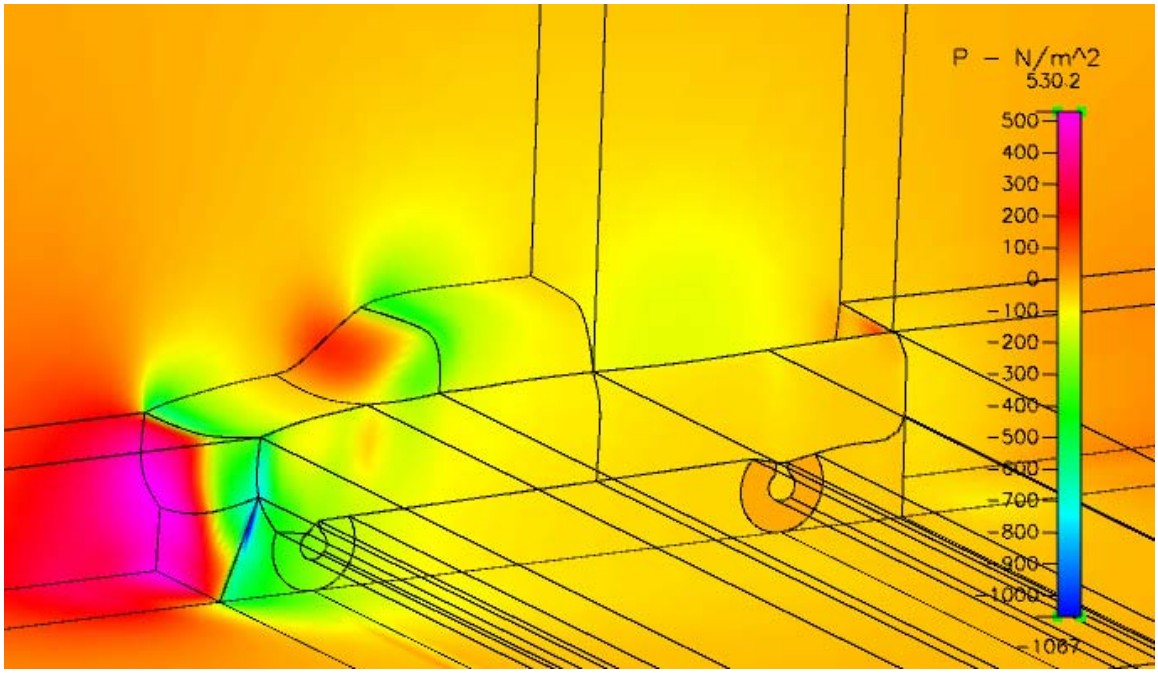

Figure 62. Air Dam and Side Skirt: Isometric Pressure Distribution

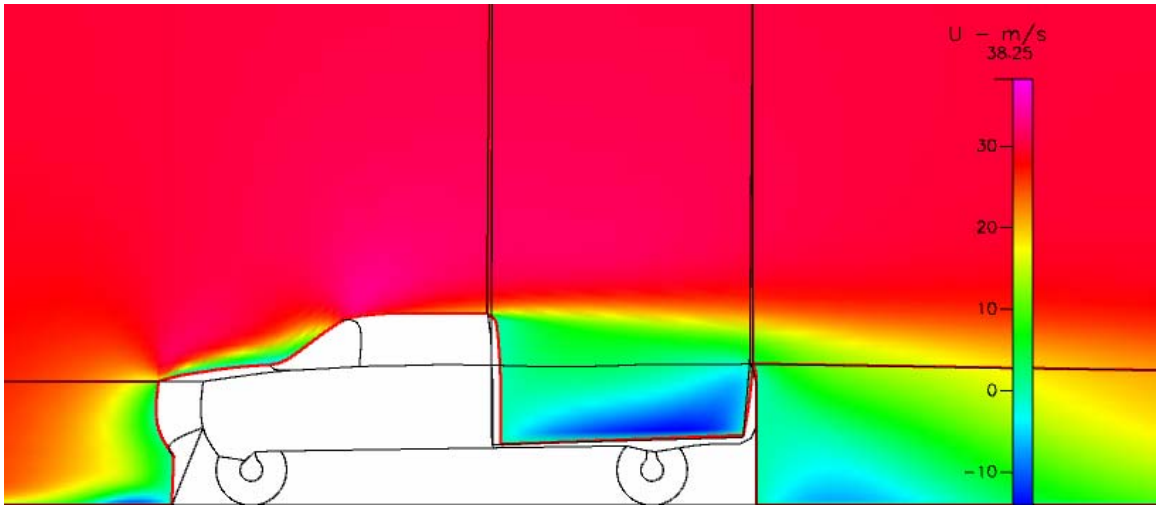

Figure 63. Air Dam and Side Skirt: X Velocity Distribution Along the Centerline

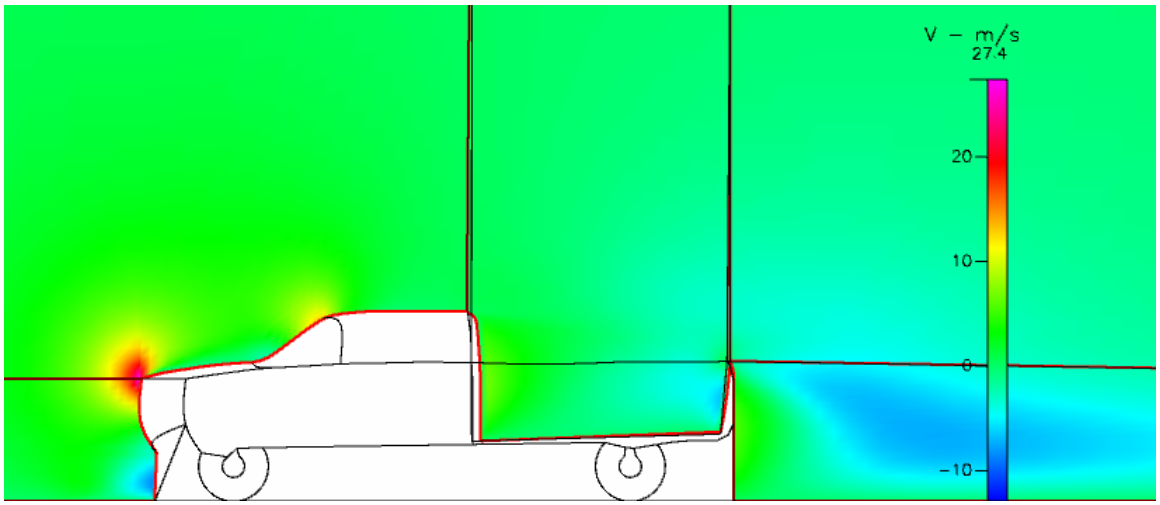

Figure 64. Air Dam and Side Skirt: Y Velocity Distribution Along the Centerline

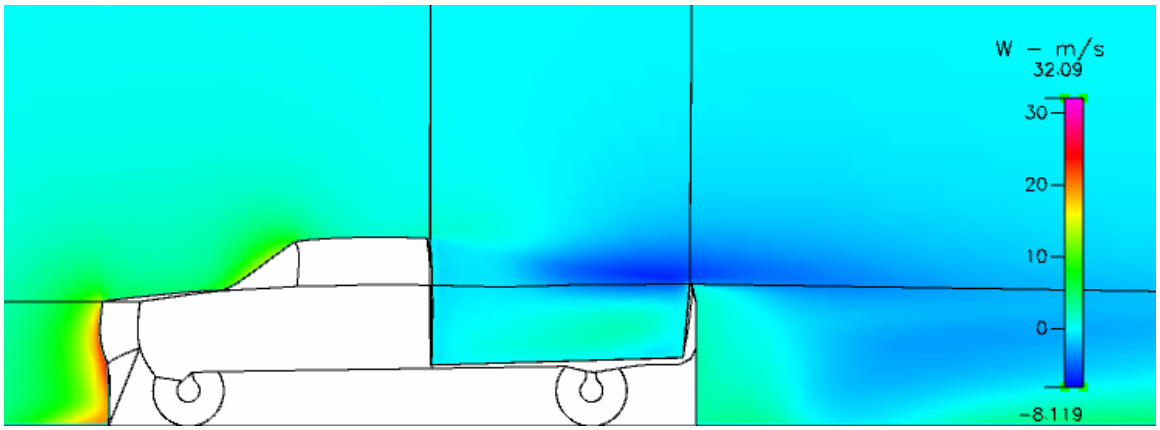

Figure 65. Air Dam and Side Skirt: Z Velocity Distribution .4 M from the Centerline

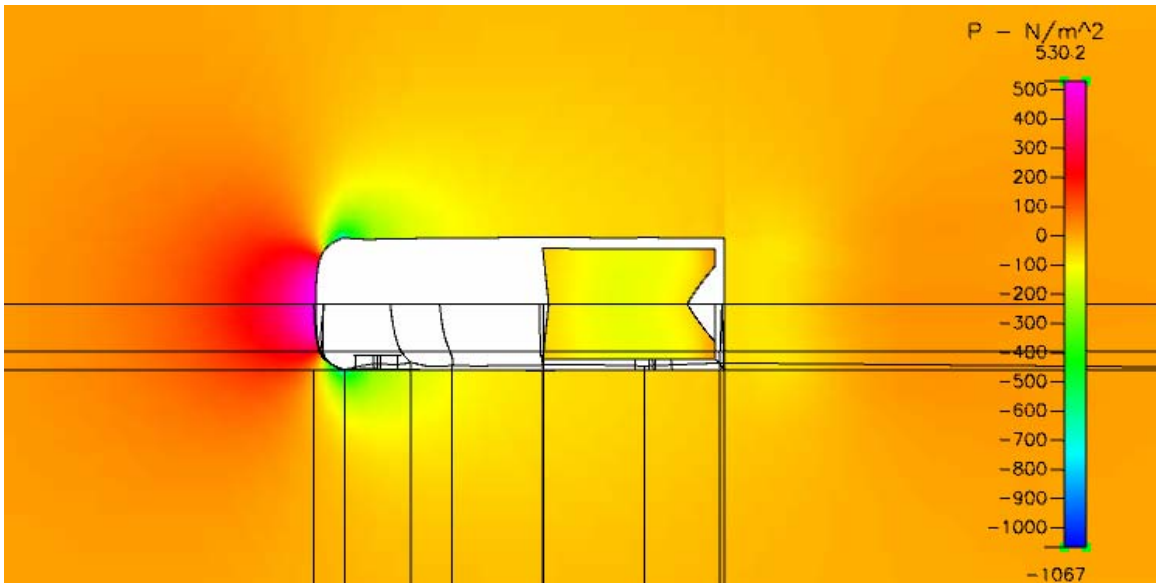

Figure 66. Air Dam and Side Skirt: Top view of Pressure Distribution

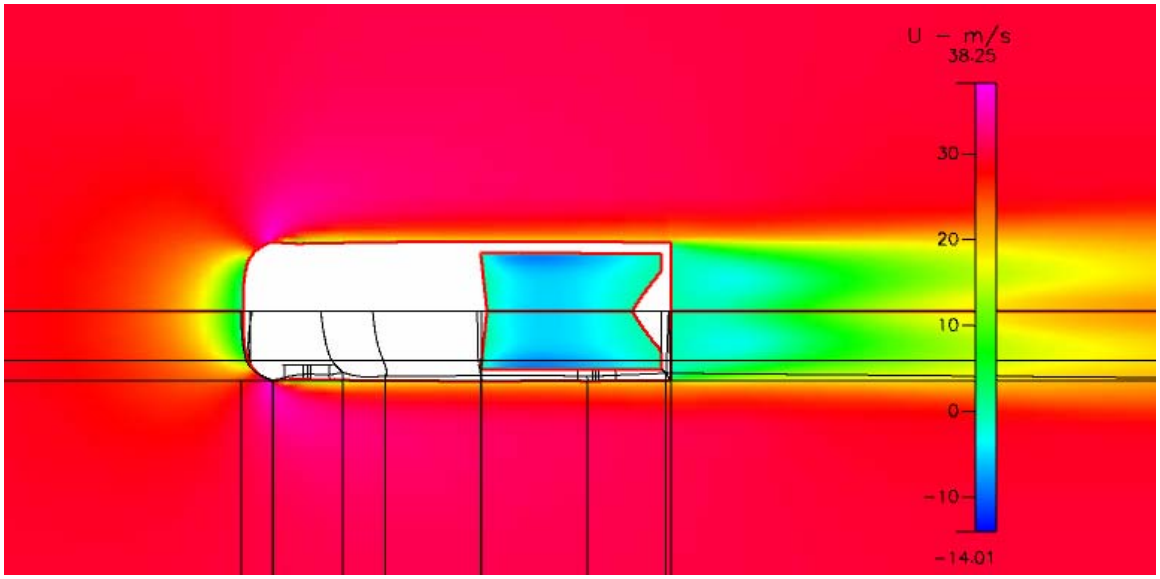

Figure 67. Air Dam and Side Skirt: Top view of X Velocity Distribution

4. 2-D Optimized Canopy

This design was developed to evaluate the effect of the optimally shaped canopy that was found in the single variable two dimensional study. It was of great interest to see whether or not the three dimensional analysis would yield similar gains.

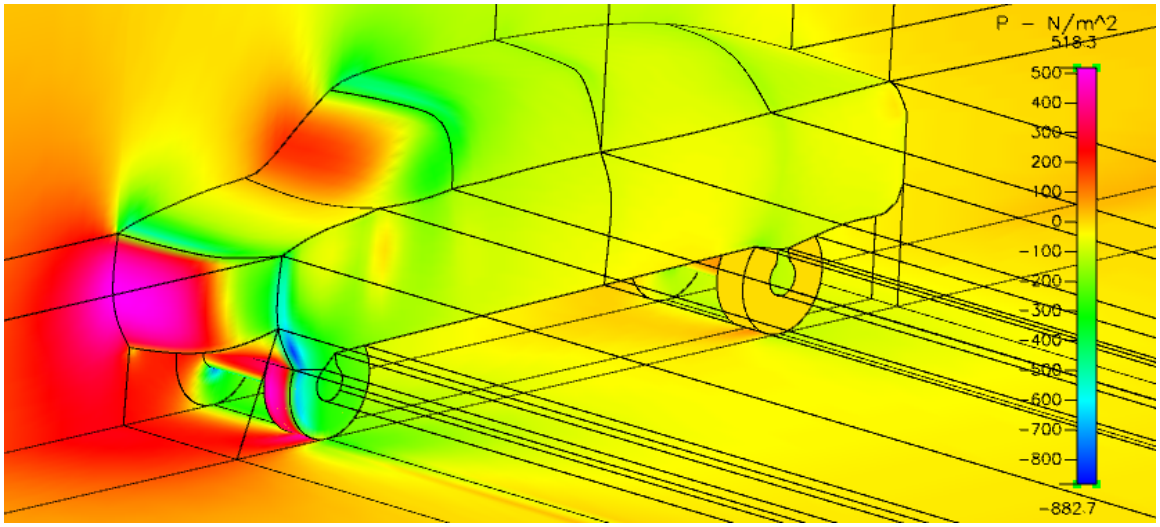

Figure 68. 2D Optimized Canopy: Isometric View of Pressure Distribution

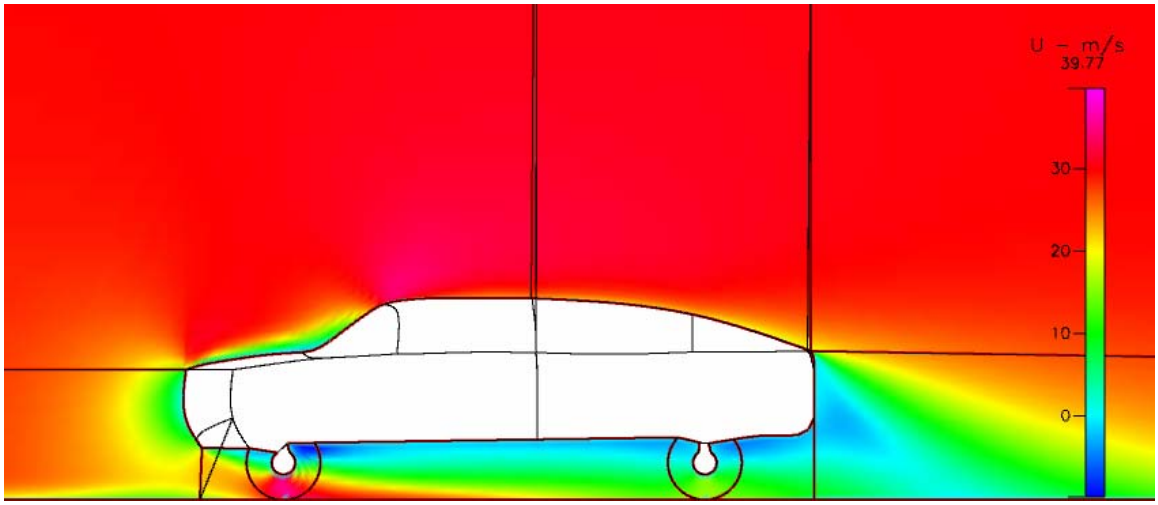

Figure 69. 2D Optimized Canopy: X Velocity Distribution Along the Centerline

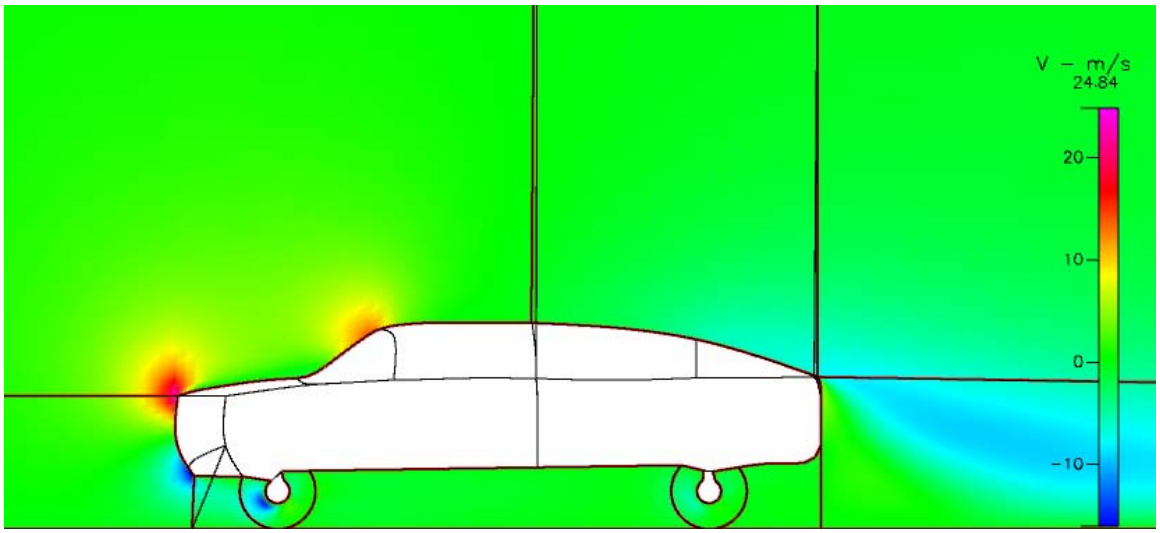

Figure 70. 2D Optimized Canopy: Y Velocity Distribution along the Centerline

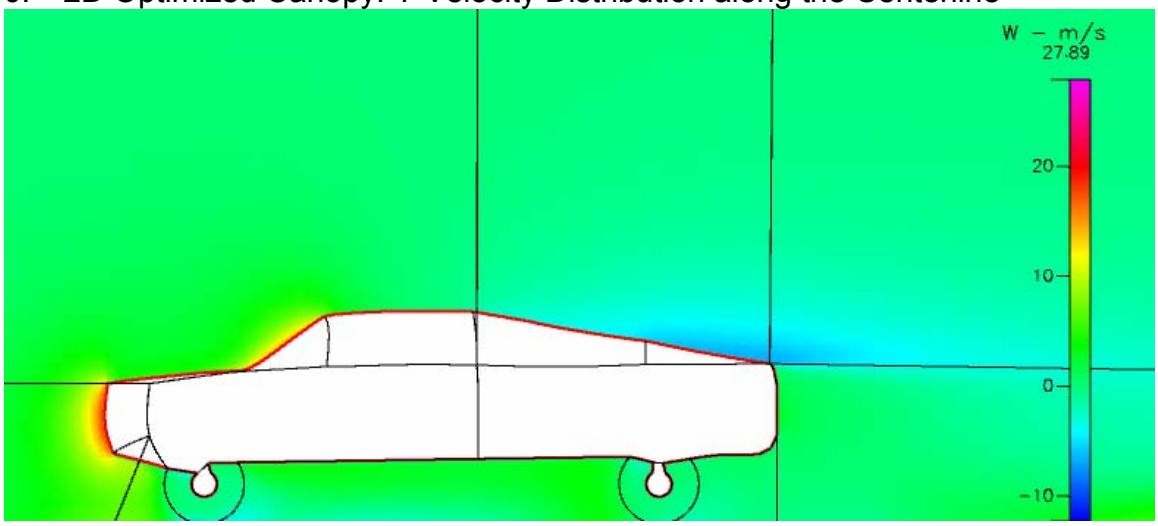

Figure 71. 2D Optimized Canopy: Z Velocity Distribution .4 M from Centerline

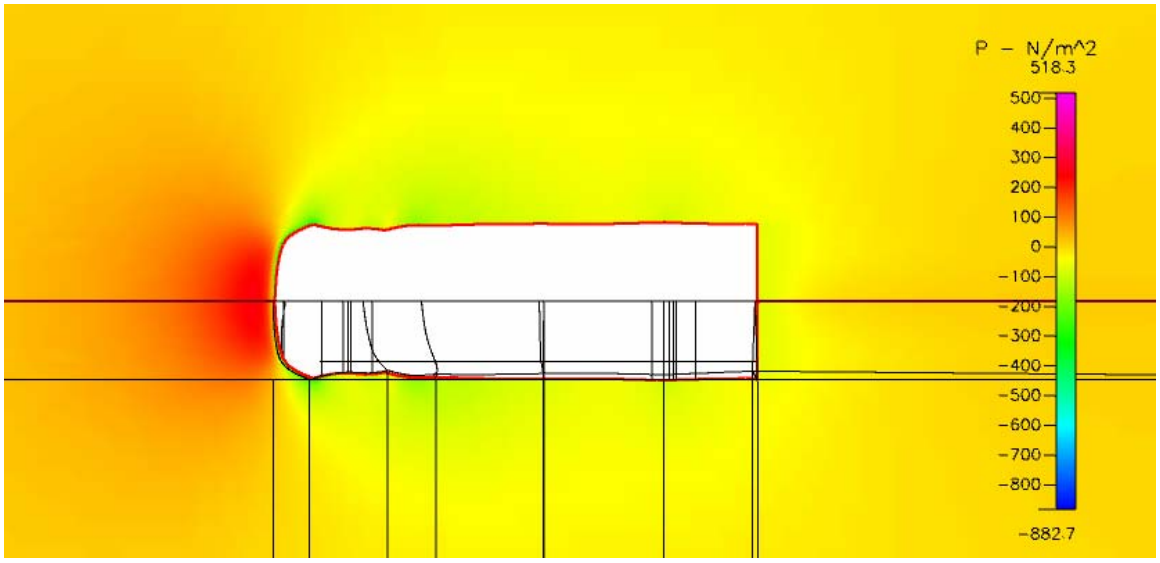

Figure 72. 2D Optimized Canopy: Top View of Pressure Distribution

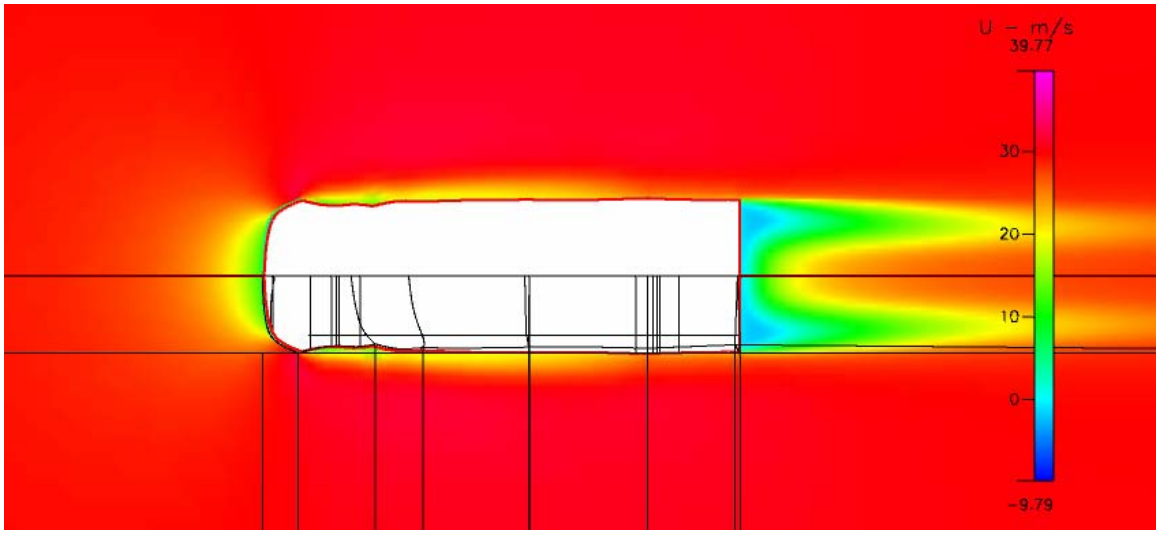

Figure 73. 2D Optimized Canopy: Top View of X Velocity Distribution

5. 2-D Optimized Canopy and Air Dam

The purpose of this design was to see the combined effects of the 2D optimized canopy and the air dam. The author was interested to see if the two effects would be negating, additive or synergistic.

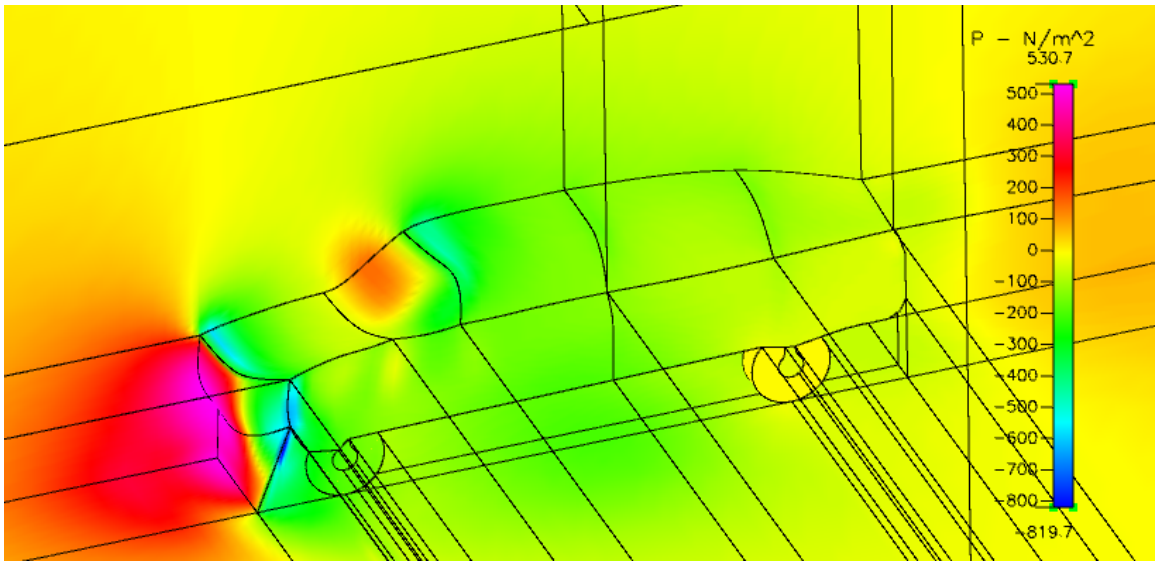

Figure 74. 2D Optimized Canopy and Air Dam: Isometric Pressure Distribution

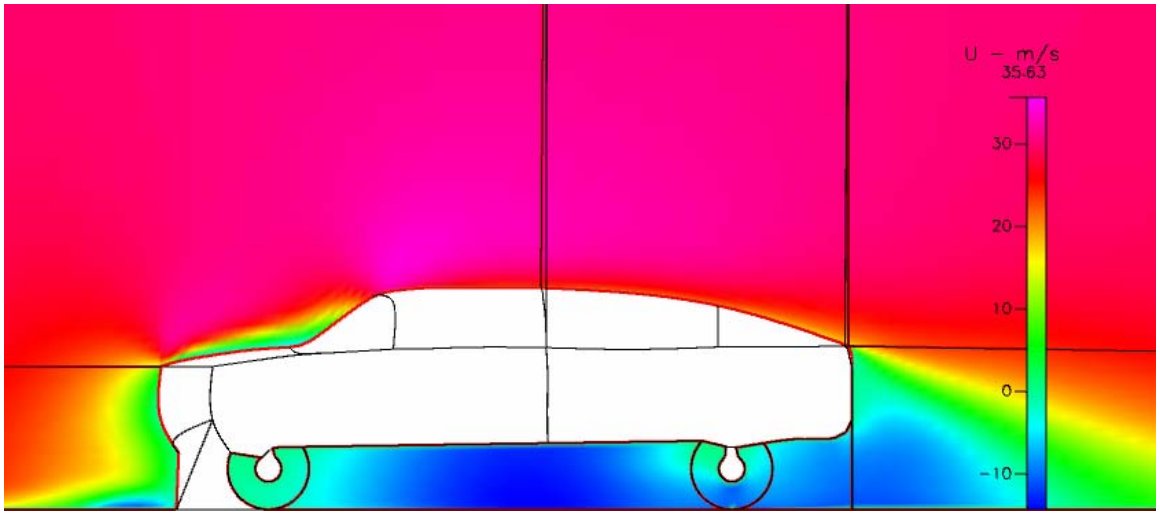

Figure 75. 2D Optimized Canopy and Air Dam: X Velocity Distribution along the Centerline

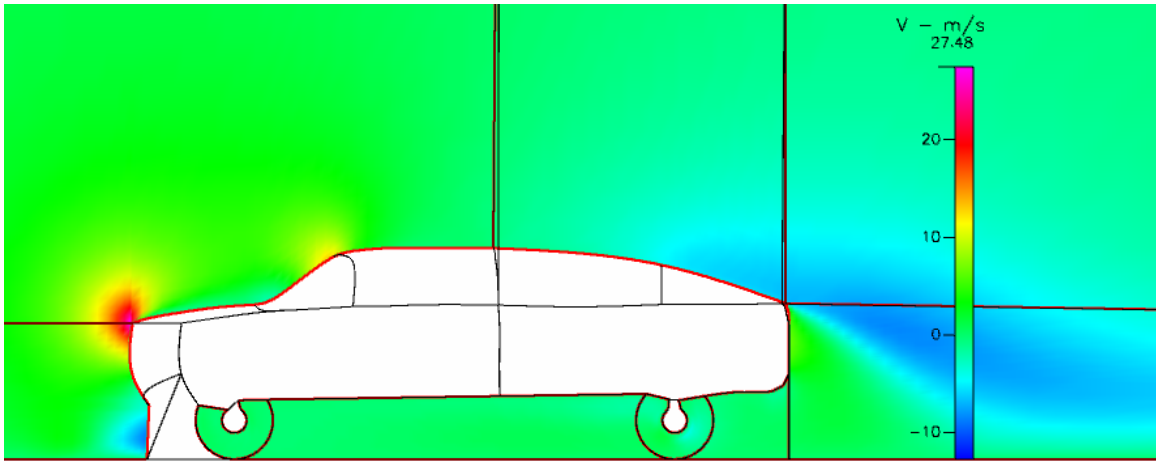

Figure 76. 2D Optimized Canopy and Air Dam: Y Velocity Distribution along the Centerline

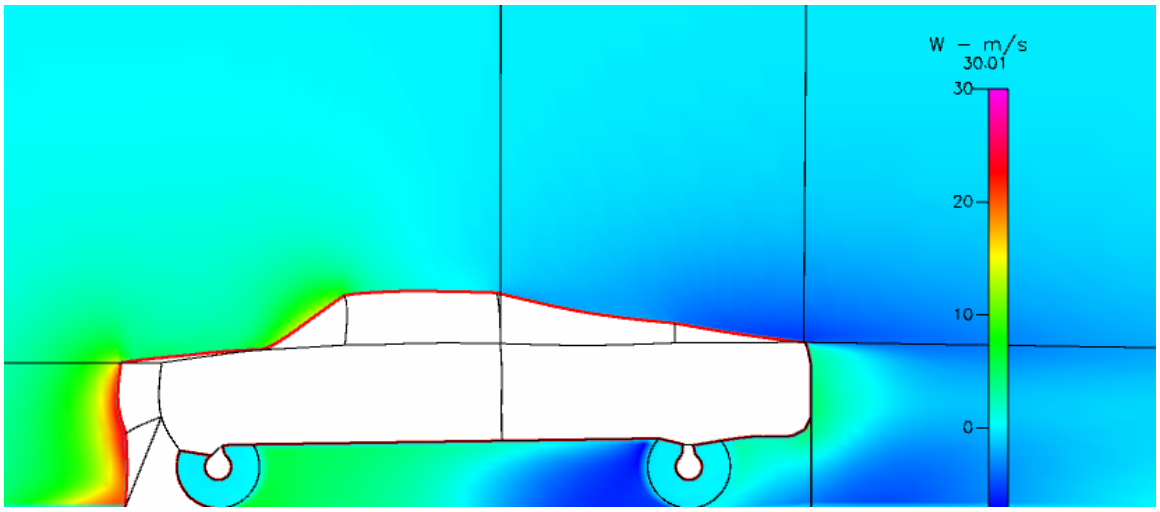

Figure 77. 2D Optimized Canopy and Air Dam: Z Velocity Distribution .4 M from Centerline

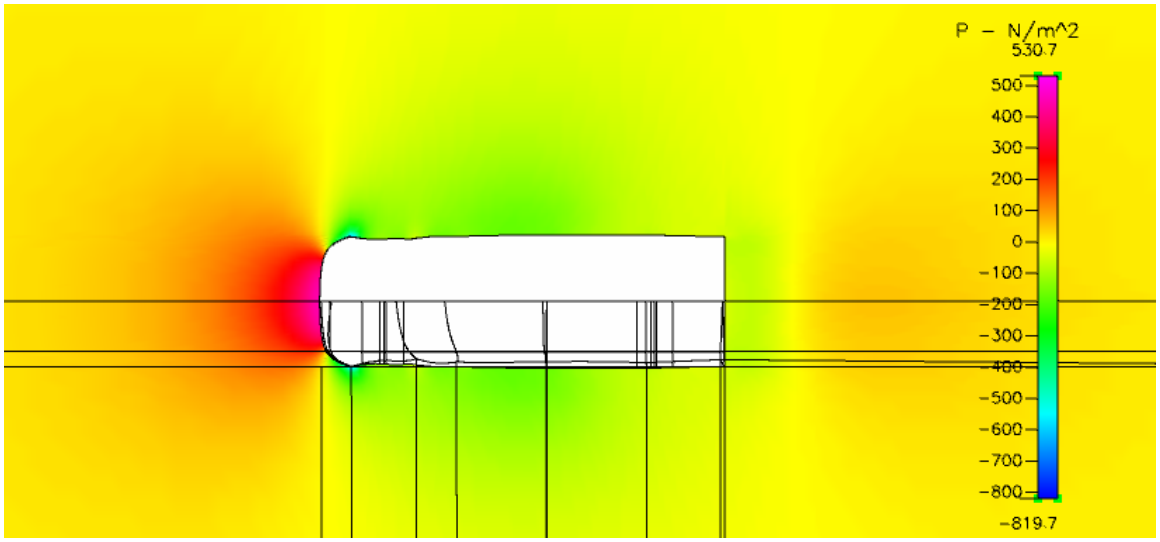

Figure 78. 2D Optimized Canopy and Air Dam: Top View of Pressure Distribution

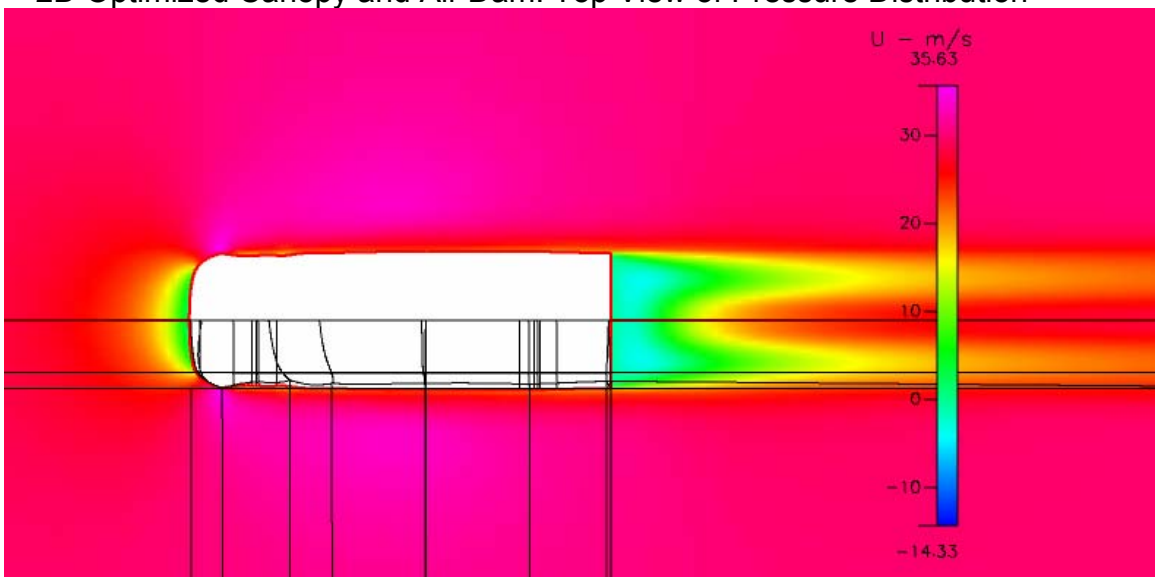

Figure 79. 2D Optimized Canopy and Air Dam: Top View of X Velocity Distribution

6. 2-D Optimized Canopy and Air Dam and Side Skirt

Along the same vein as the previous design it was of interest to see whether the addition of side skirts would improve the 2D optimized canopy and air dam only design.

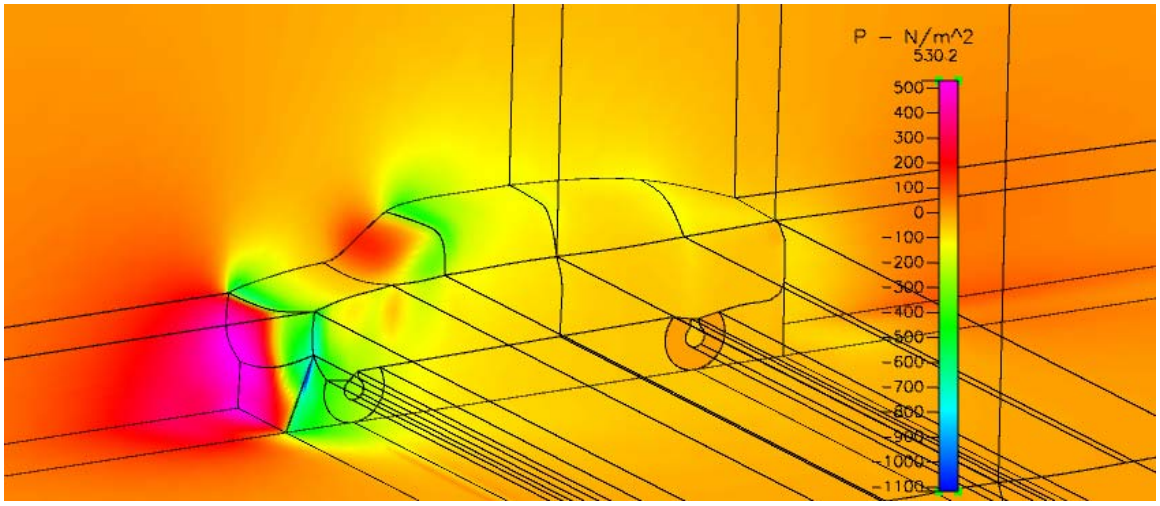

Figure 80. 2D Optimized Canopy, Air Dam, Side Skirt: Isometric View of Pressure Distribution

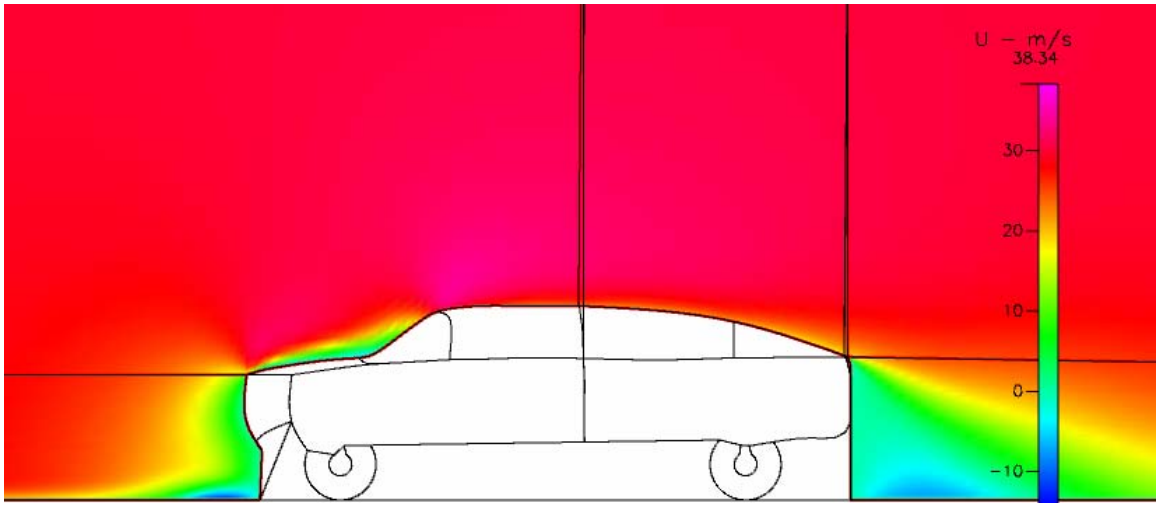

Figure 81. 2D Optimized Canopy, Air Dam, Side Skirt: X Velocity Distribution along the Centerline

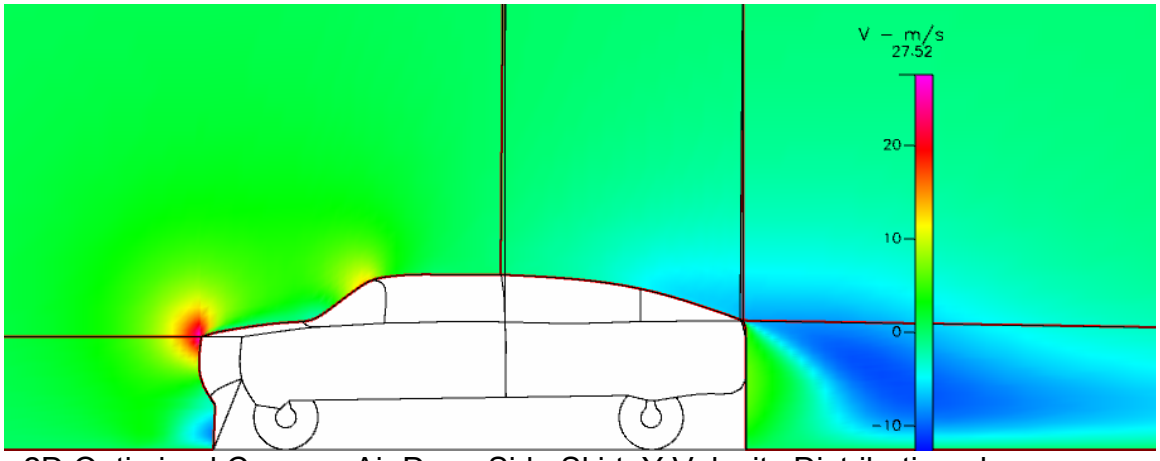

Figure 82. 2D Optimized Canopy, Air Dam, Side Skirt: Y Velocity Distribution along the Centerline

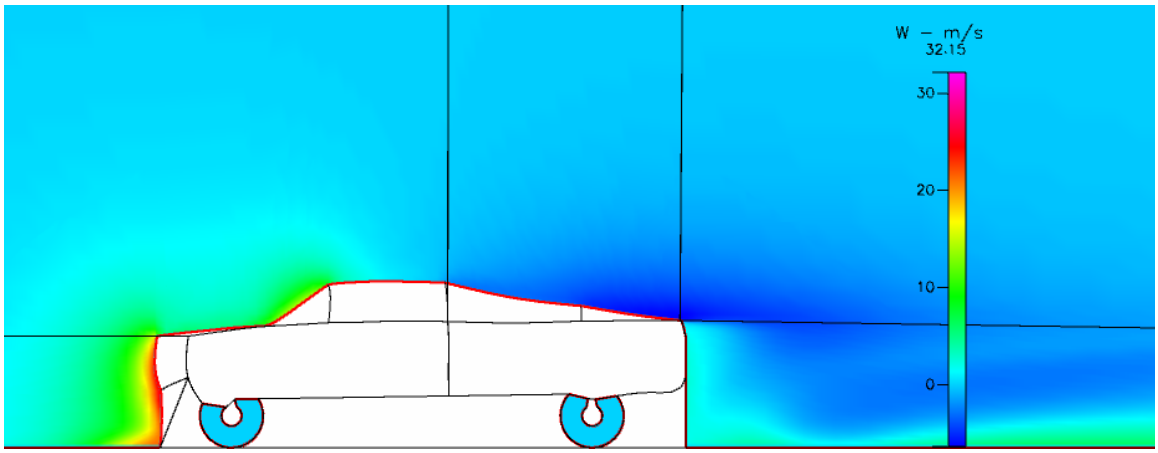

Figure 83. 2D Optimized Canopy, Air Dam, Side Skirt: Z Velocity Distribution .4 M from Centerline

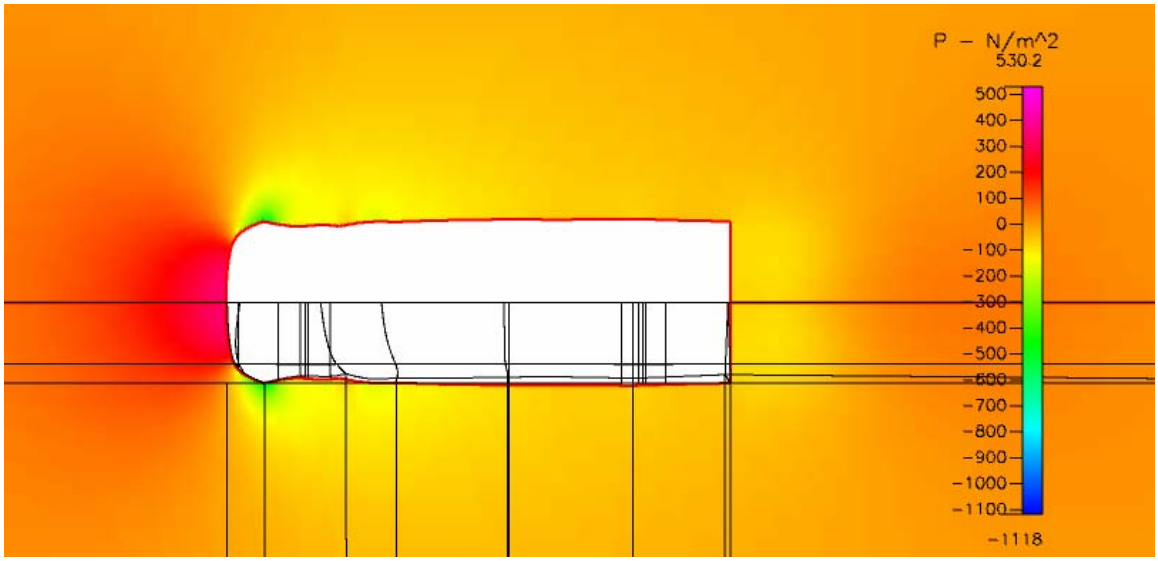

Figure 84. 2D Optimized Canopy, Air Dam, Side Skirt: Top View of Pressure Distribution

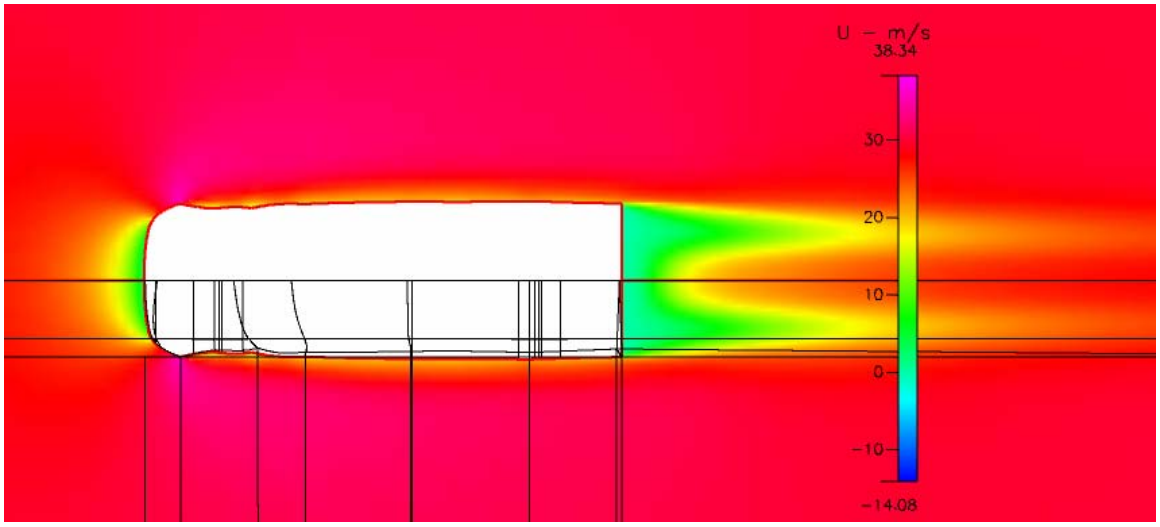

Figure 85. 2D Optimized Canopy, Air Dam, Side Skirt: Top View of X Velocity Distribution

7. Tonneau Cover

The next design of interest is one that is commercially available. The tonneau cover is quickly becoming one of the more popular accessories for light trucks. The effect of the tonneau cover was relevant to this research so a model was built where the pickup bed was sealed with a flat boundary condition similar to a tonneau cover. The axles and wheels remain exposed without an air dam or side skirts for this model.

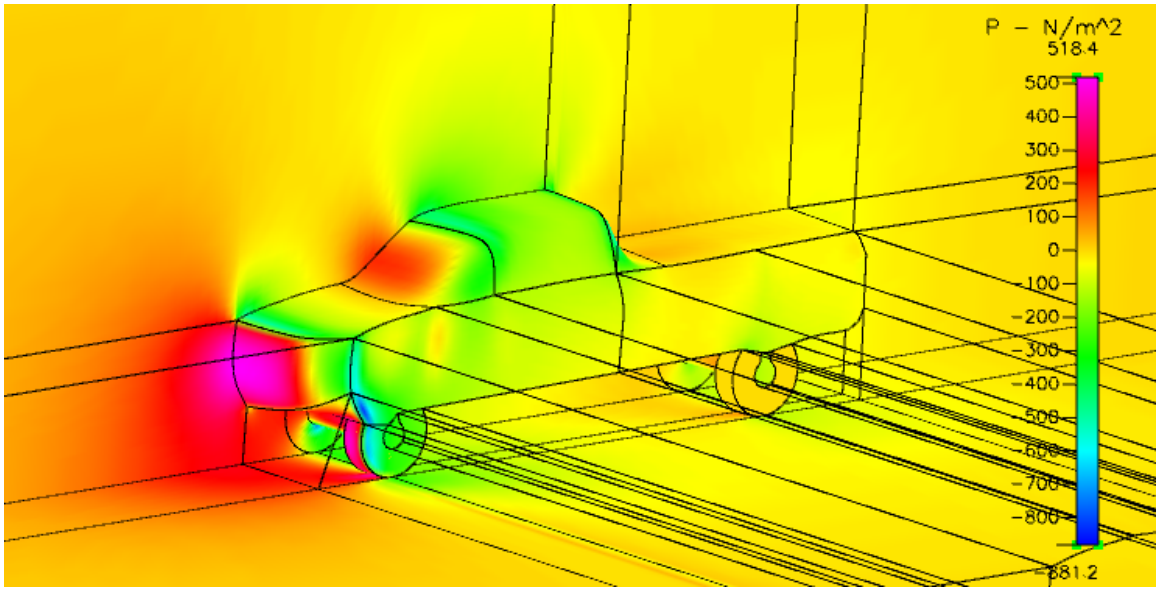

Figure 86. Tonneau Cover: Isometric View of Pressure Distribution

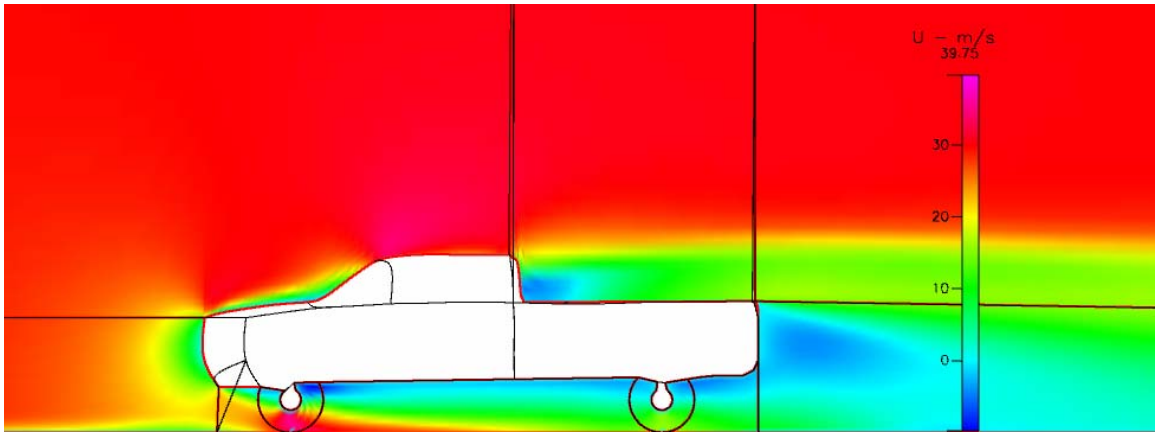

Figure 87. Tonneau Cover: X Velocity Distribution along Centerline

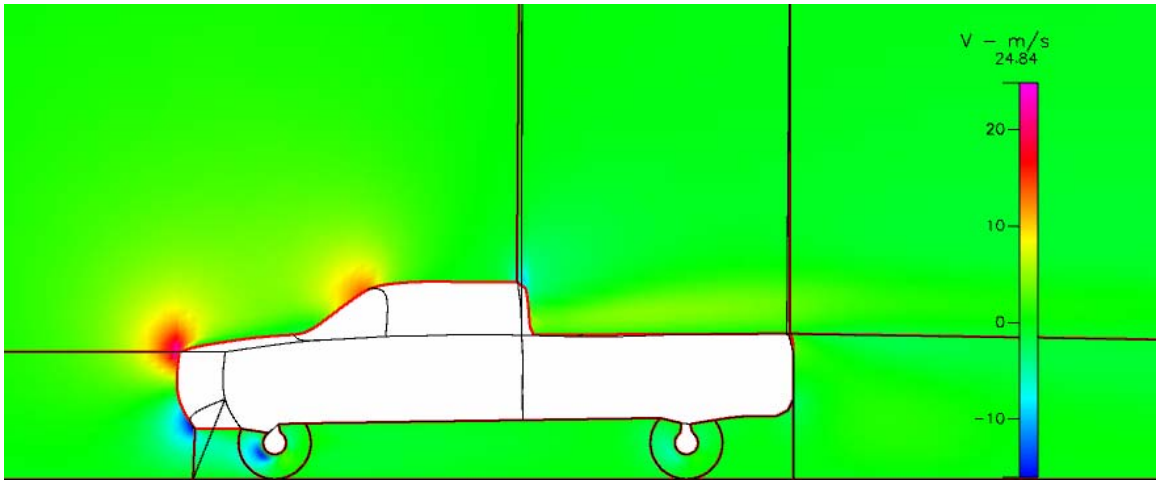

Figure 88. Tonneau Cover: Y Velocity Distribution along Centerline

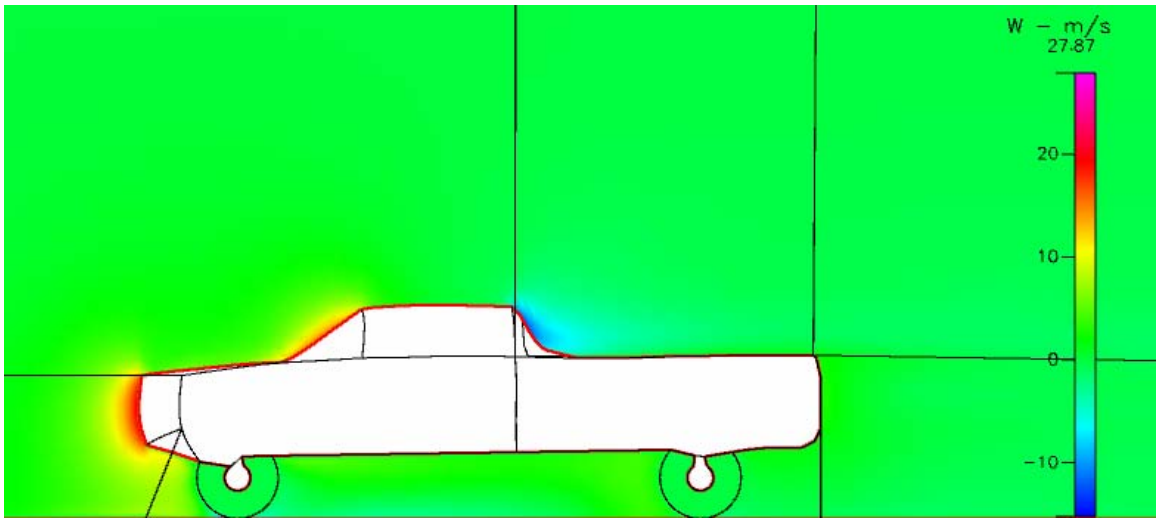

Figure 89. Tonneau Cover: Z Velocity Distribtion .4 M from Centerline

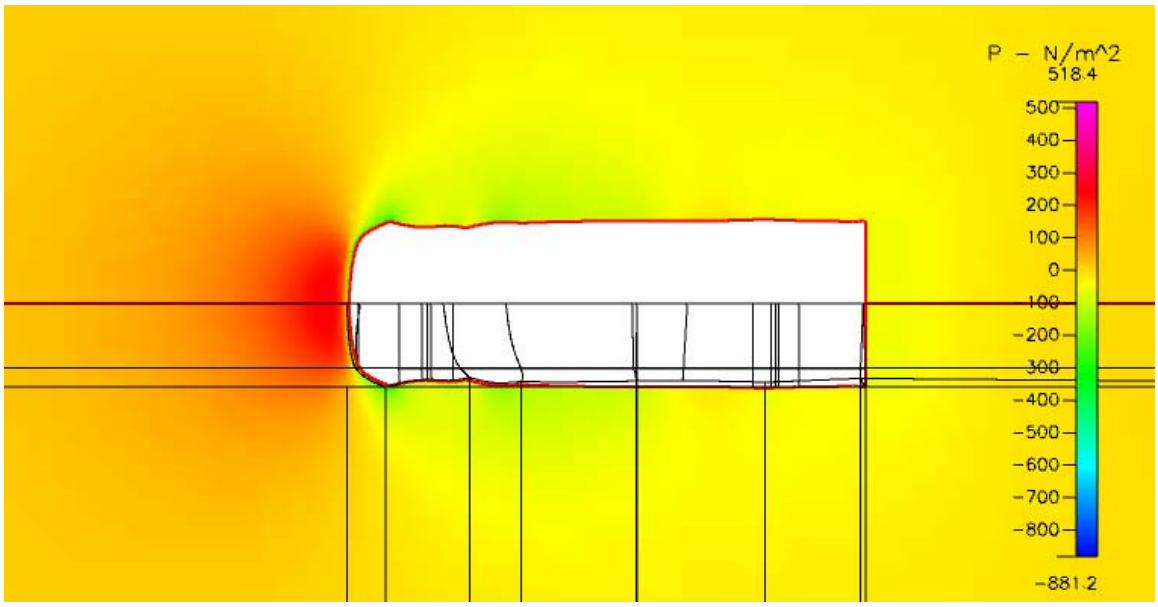

Figure 90. Tonneau Cover - Pressure Distribution Top View

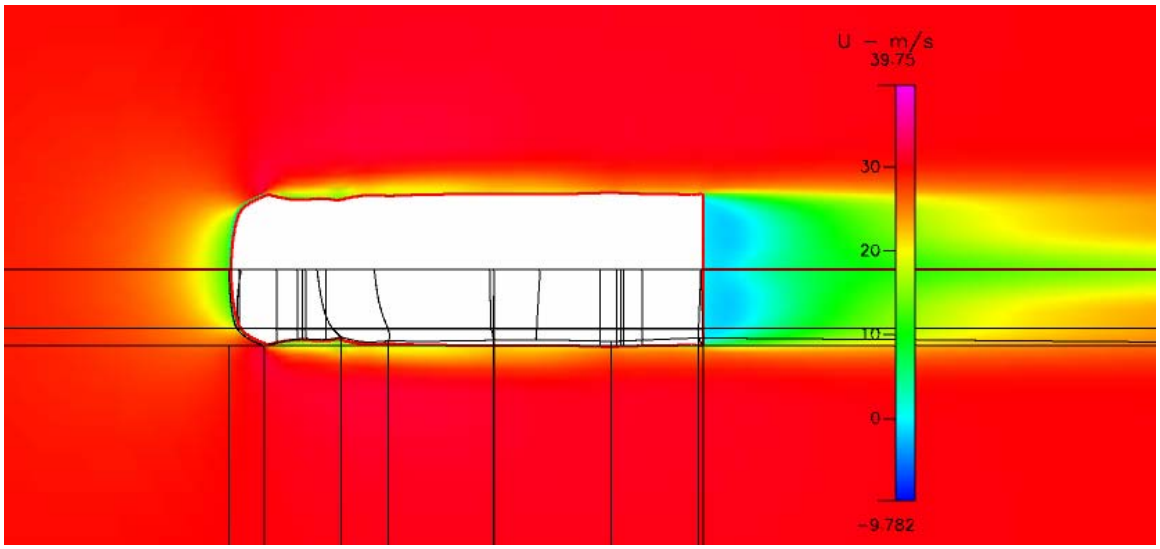

Figure 91. Tonneau cover - X Velocity Distribution Top View

8. Traditional Canopy

The traditional canopy was of interest because it is also commercially available. It also has the desirable property of enclosing a large volume of bed space. Ultimately, this is one of the most important designs to evaluate because it is the one most commonly used and it provides a good comparison for the improvements made by optimally shaped canopies.

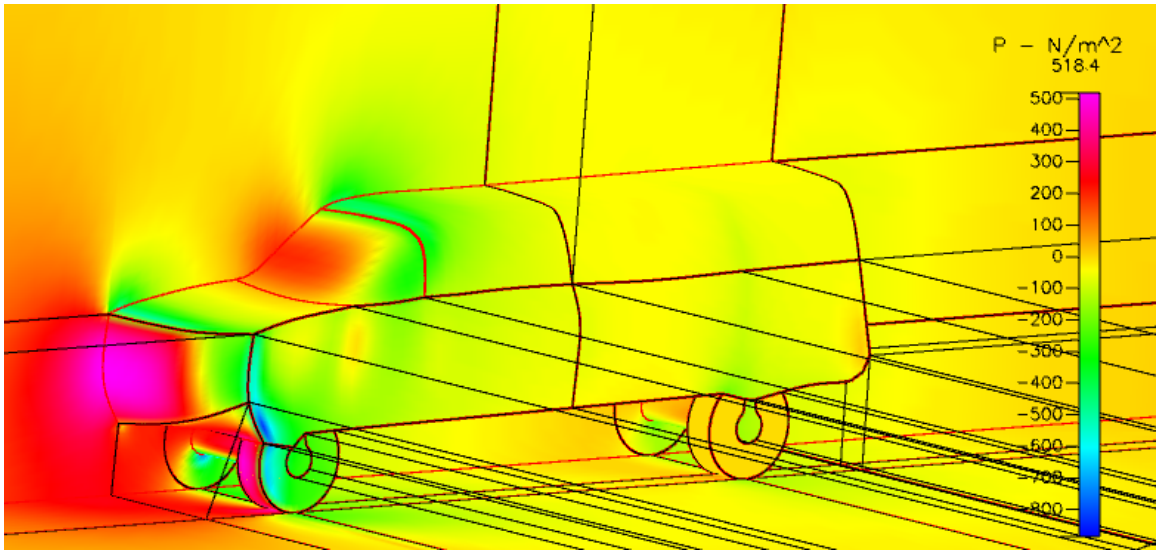

Figure 92. Traditional Canopy: Frontal Isometric View of Pressure Distribution

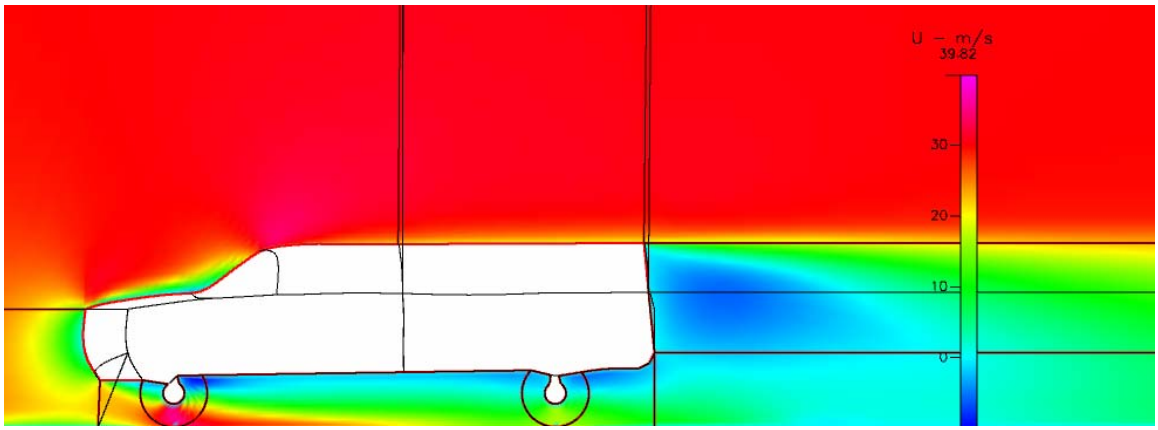

Figure 93. Traditional Canopy: X Velocity Distribution along Centerline

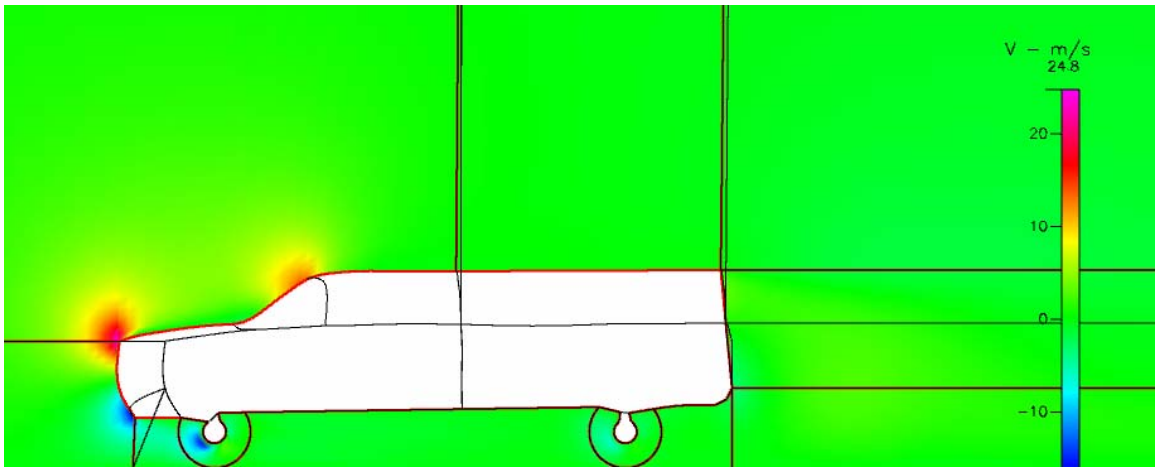

Figure 94. Traditional Canopy: Y Velocity Distribution along Centerline

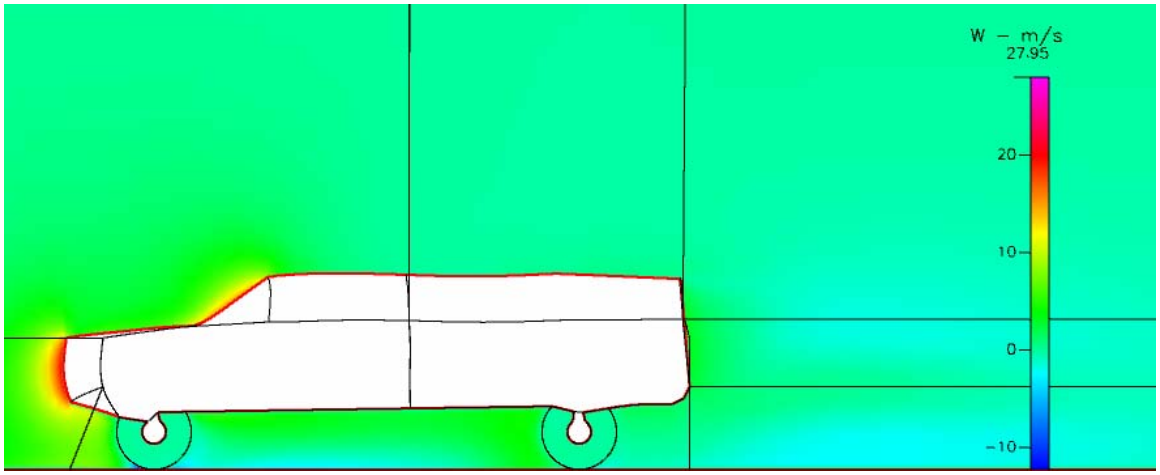

Figure 95. Traditional Canopy: Z Velocity Distribution .4 M from Centerline

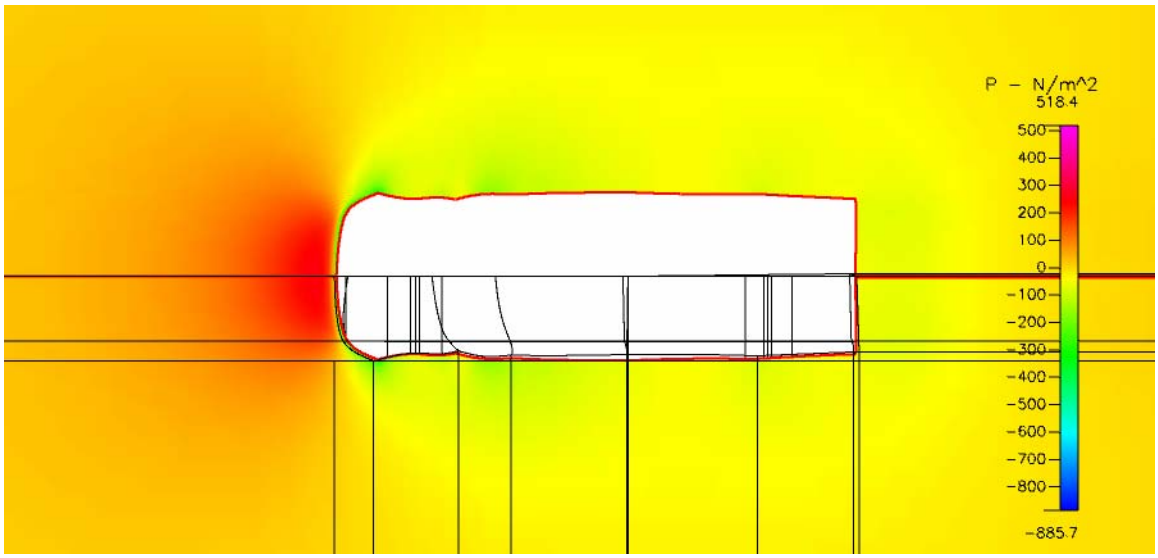

Figure 96. Traditional Canopy: Top View of Pressure Distribution

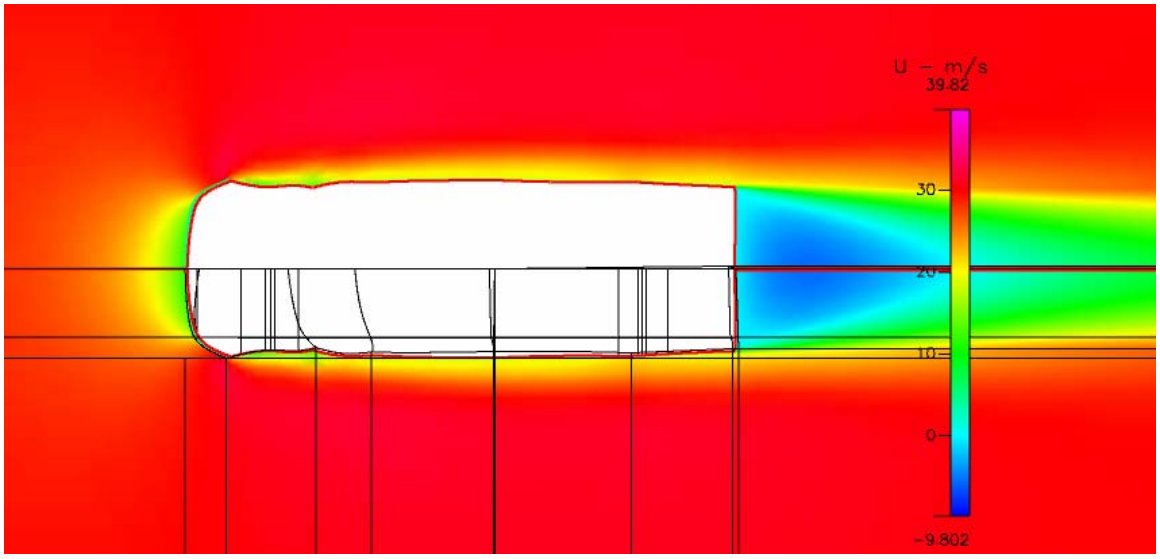

Figure 97. Traditional Canopy: Top View of X Velocity Distribution

9. Tonneau Cover, Air Dam and Side Skirt

This model was developed to examine the effects of air dam and side skirts on the commercially available tonneau covers.

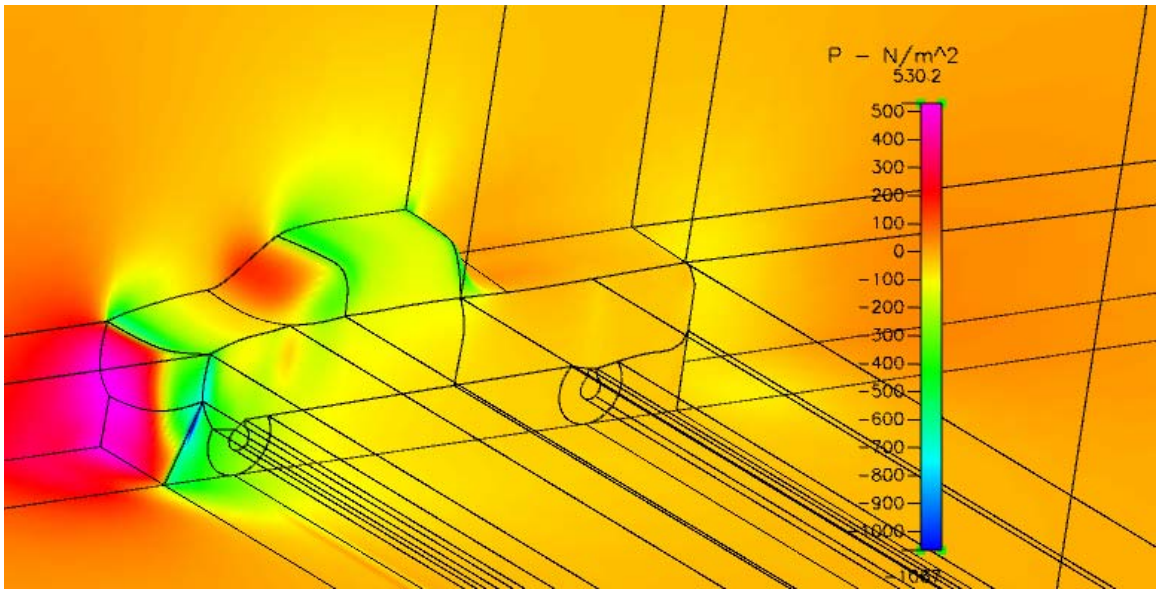

Figure 98. Tonneau Cover, Air Dam, Side Skirts: Isometric View of Pressure Distribution

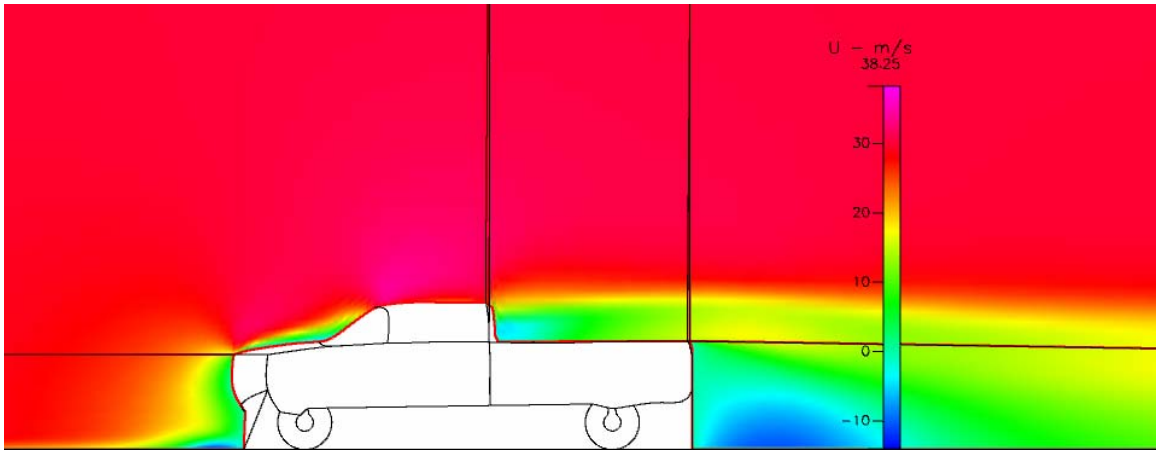

Figure 99. Tonneau Cover, Air Dam, Side Skirts: X Velocity Distribution along Centerline

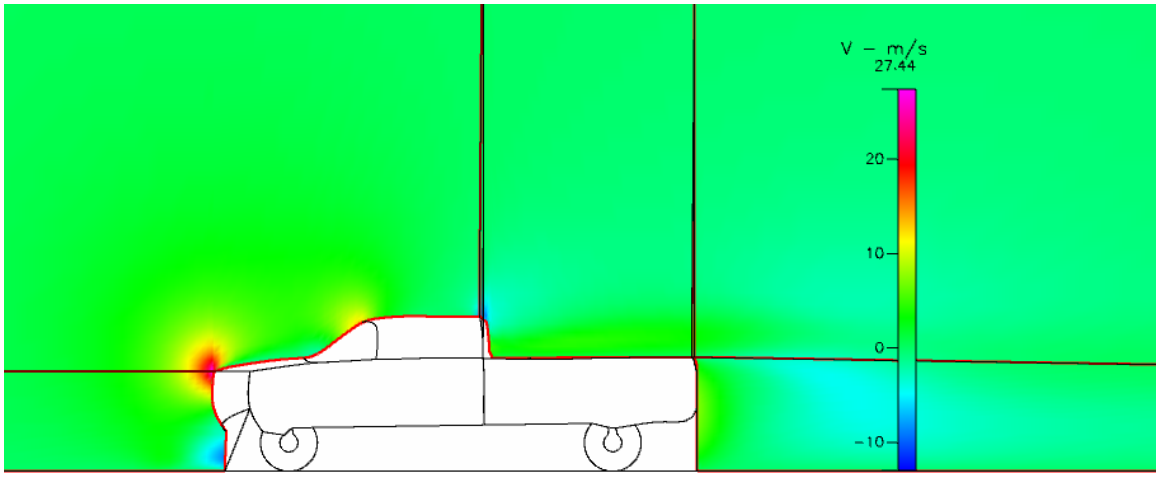

Figure 100. Tonneau Cover, Air Dam, and Side Skirts: Y Velocity Distribution along Centerline

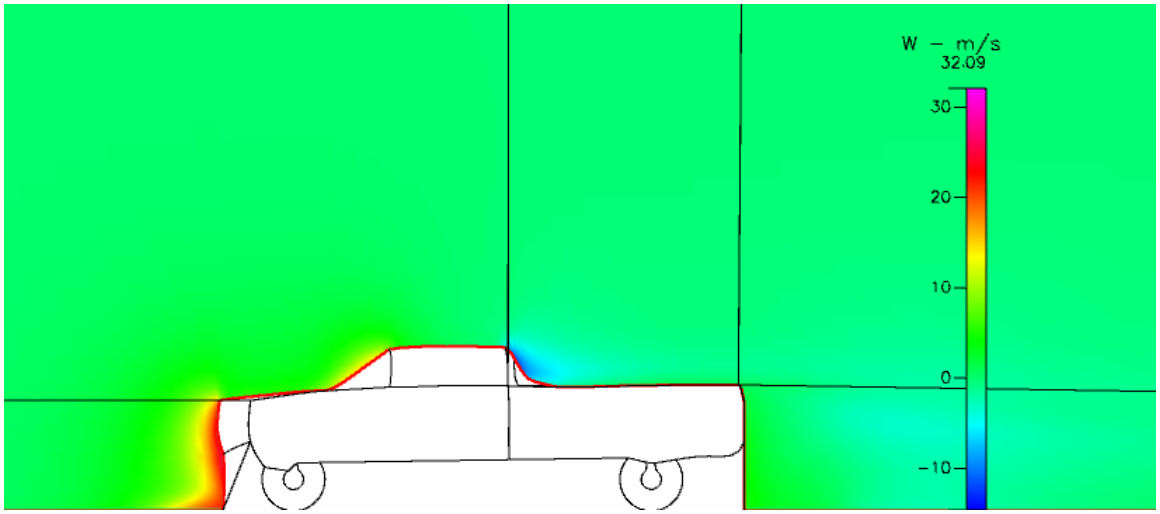

Figure 101. Tonneau Cover, Air Dam, Side Skirts: Z Velocity Distribution at .4 M from Centerline

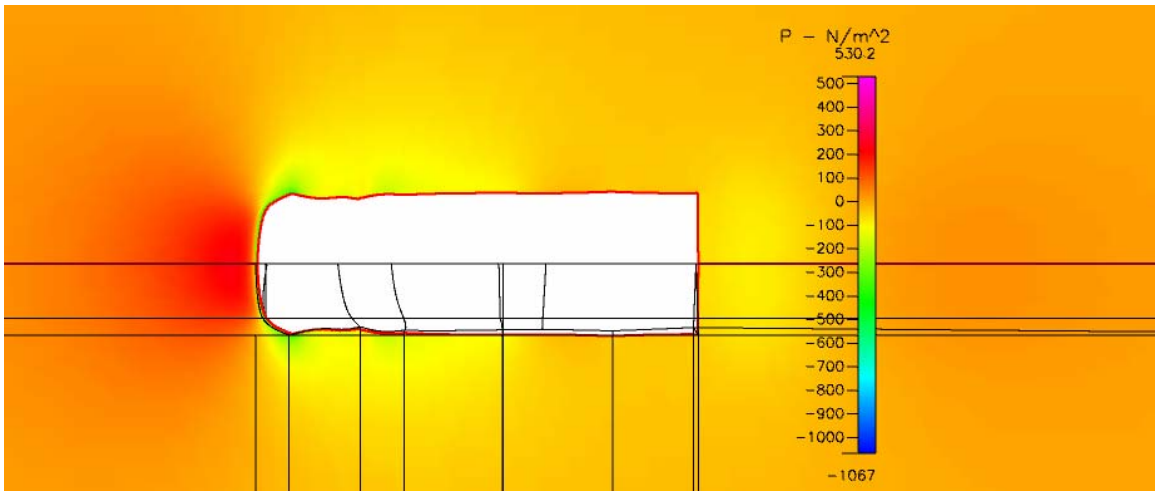

Figure 102. Tonneau Cover, Air Dam, Side Skirts: Top view of Pressure Distribution

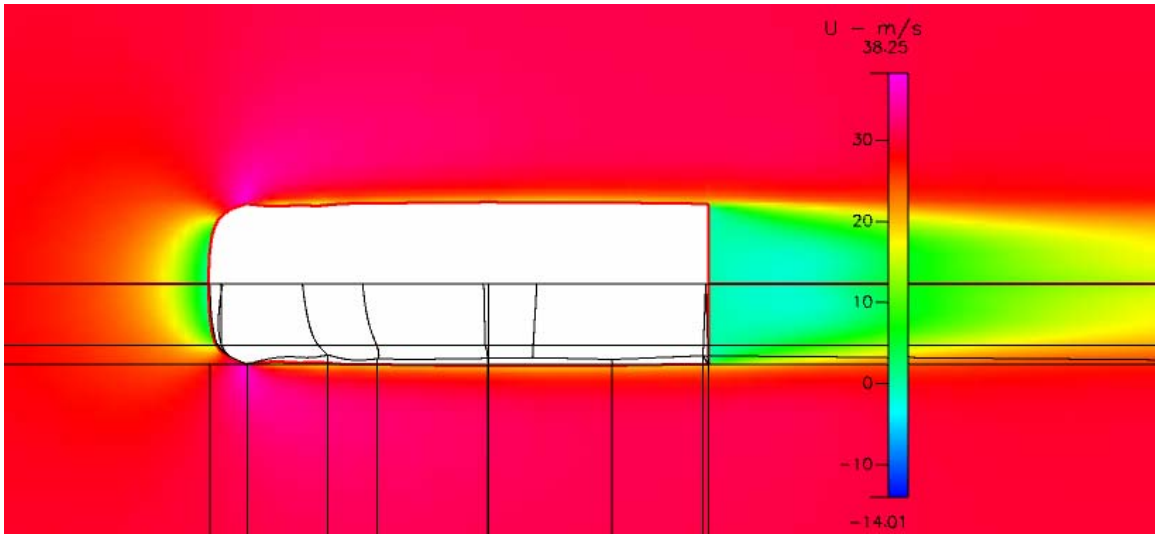

Figure 103. Tonneau Cover, Air Dam, Side Skirts: Top View of X Velocity Distribution

10. Traditional Canopy, Air Dam and Side Skirt

The next design of interest is similar to the previous with the exception that the Tonneau Cover is swapped for a traditional canopy.

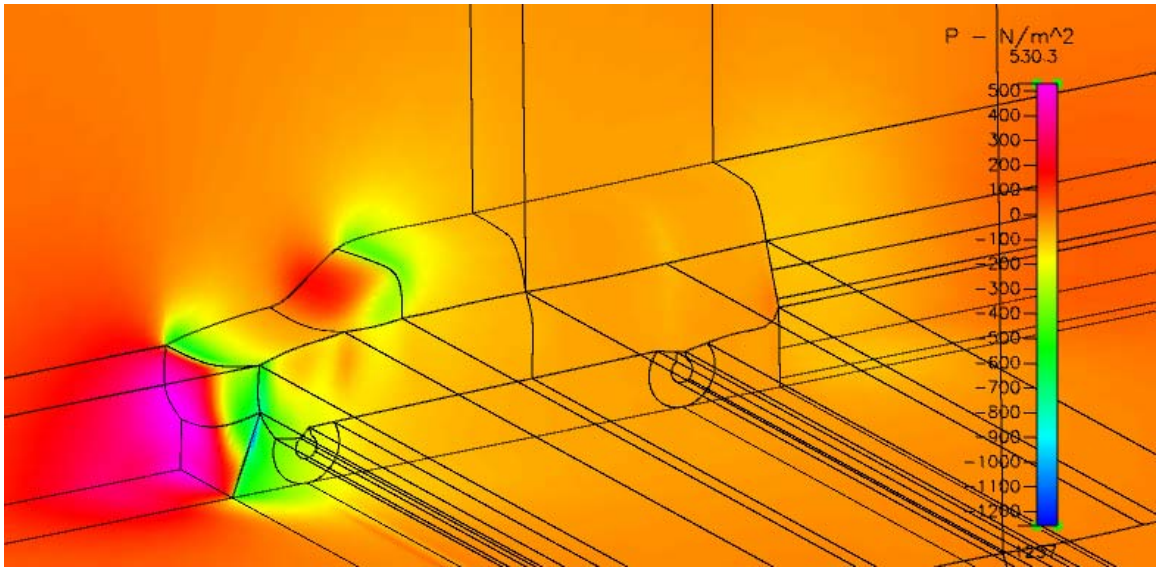

Figure 104. Traditional Canopy, Air Dam and Side Skirts: Isometric View of Pressure Distribution

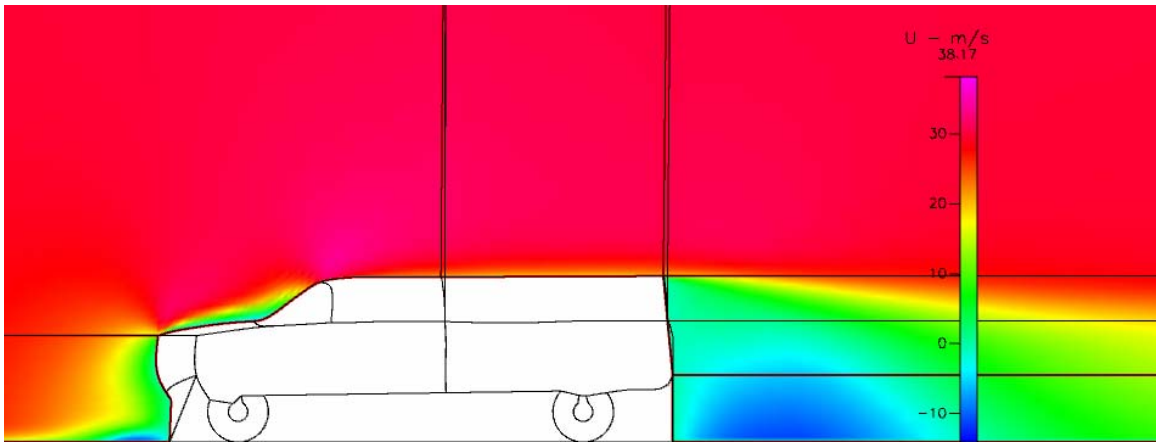

Figure 105. Traditional Canopy, Air Dam, Side Skirts: X Velocity Distribution along Centerline

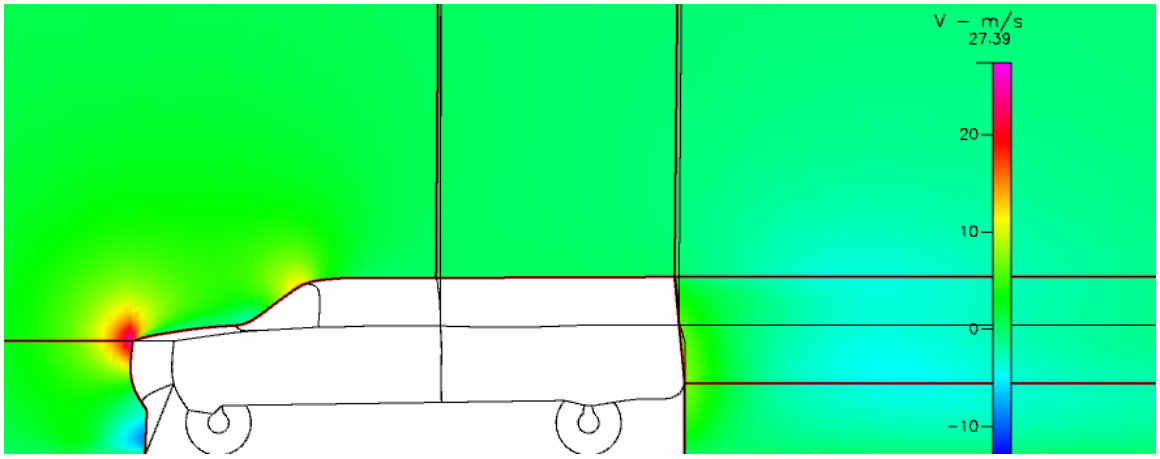

Figure 106. Traditional Canopy, air Dam, Side Skirts: Y Velocity Distribution along Centerline

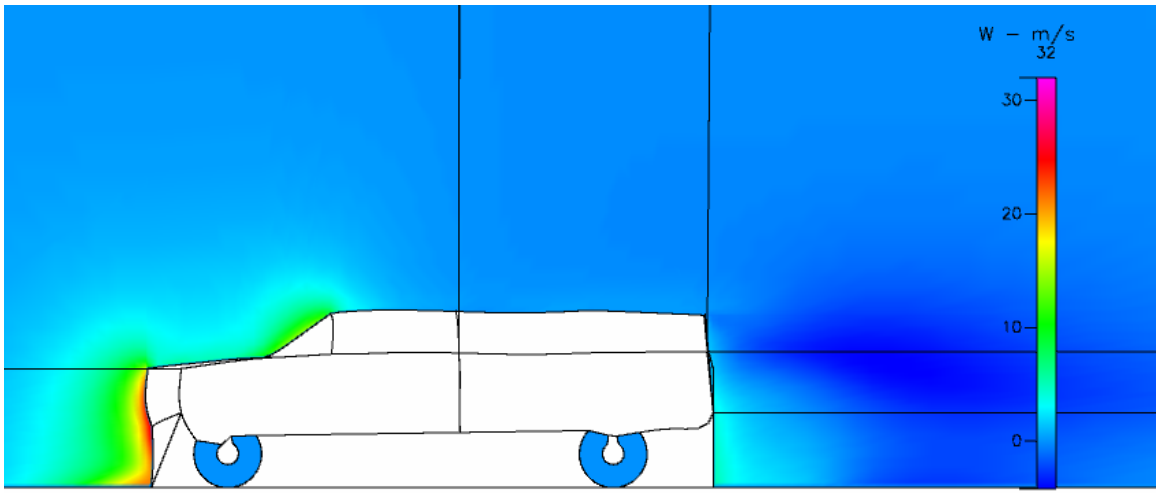

Figure 107. Traditional Canopy, Air Dam, Side Skirts: Z Velocity Distribution .4 M from Centerline

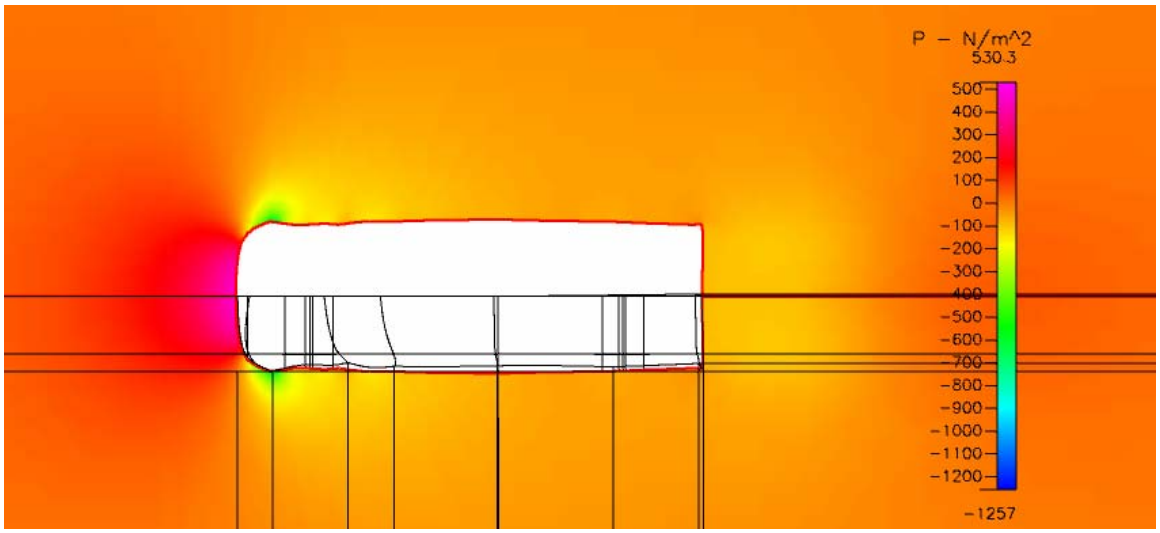

Figure 108. Traditional Canopy, Air Dam, Side Skirts: Top View of Pressure Distribution

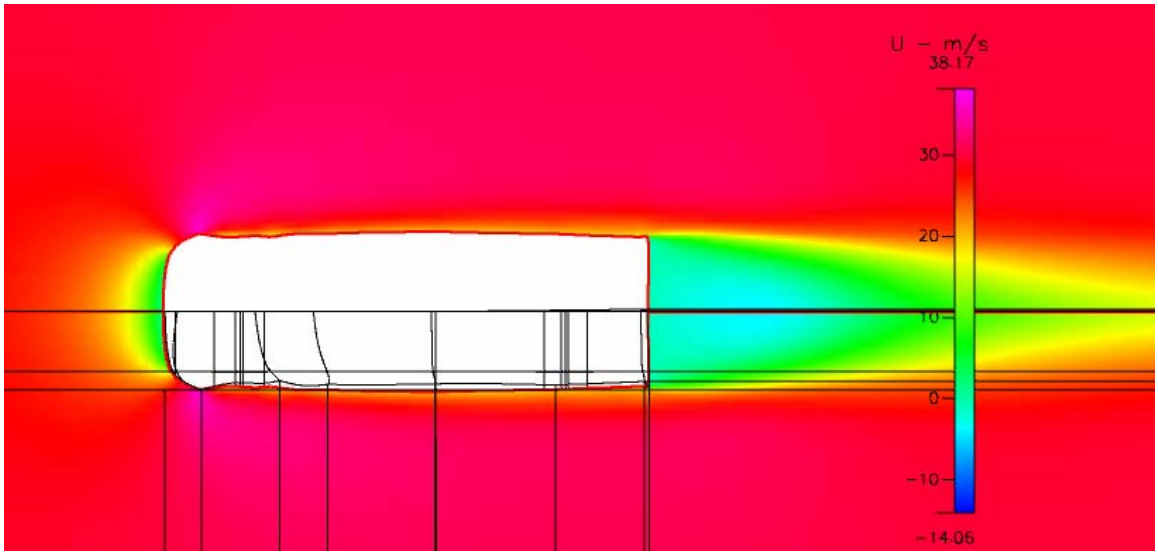

Figure 109. Traditional Canopy, Air Dam, Side Skirts: Top View of X Velocity Distribution

11. 3D Curved Canopy

After some careful evaluation it was determine that the 2D optimal canopy model would not prove very flexible in generating many designs for the optimization in the next section. This was because the 2D optimal canopy was constrained to finishing flush with the tailgate. In order to increase the domain of the optimization, the author chose to create a new baseline for the optimization model called the 3D Curved Canopy. This design allows the optimizer to vary

the height of the chord at the tail gate in addition to the chord midway up the canopy. Ultimately, many more interesting designs are possible for the optimizer with the addition of only two design variables. The baseline for this whole process is presented as the 3D Curved Canopy below:

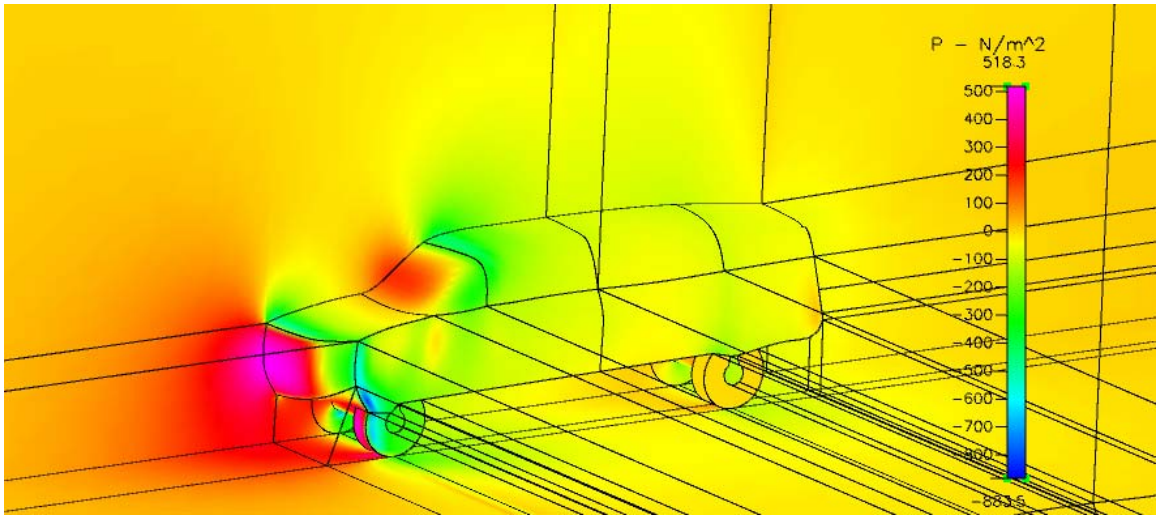

Figure 110. 3D Curved Canopy: Isometric View of Pressure Distribution

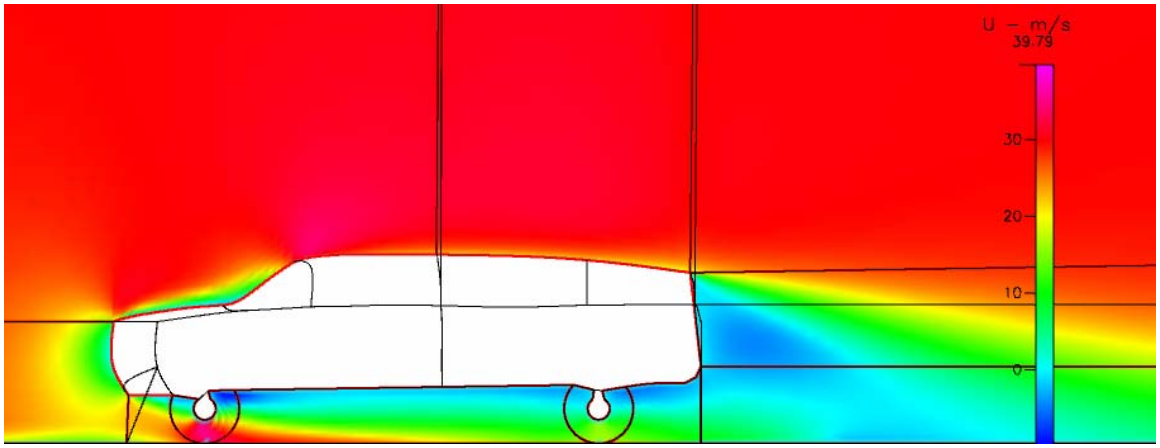

Figure 111. 3D Curved Canopy: X Velocity Distribution along Centerline

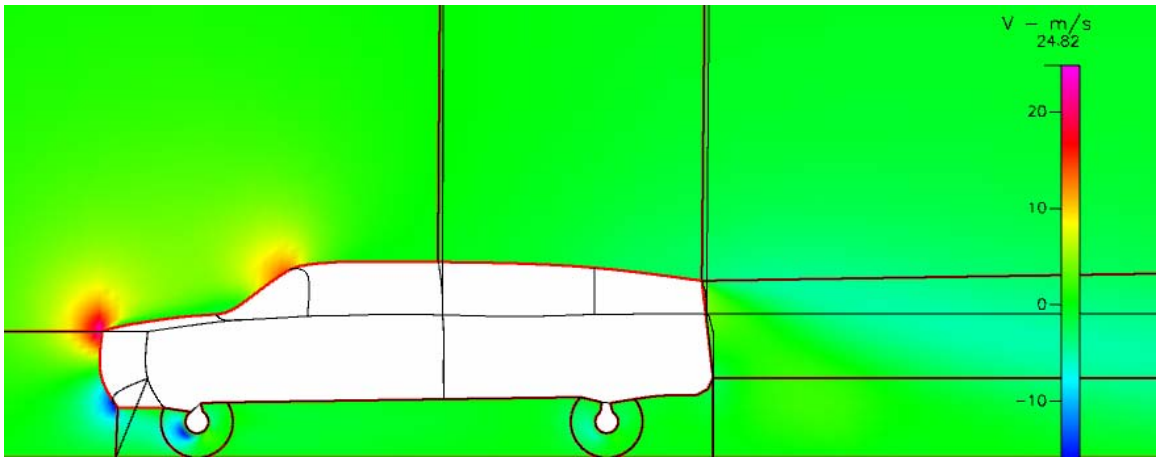

Figure 112. 3D Curved Canopy: Y Velocity Distribution along Centerline

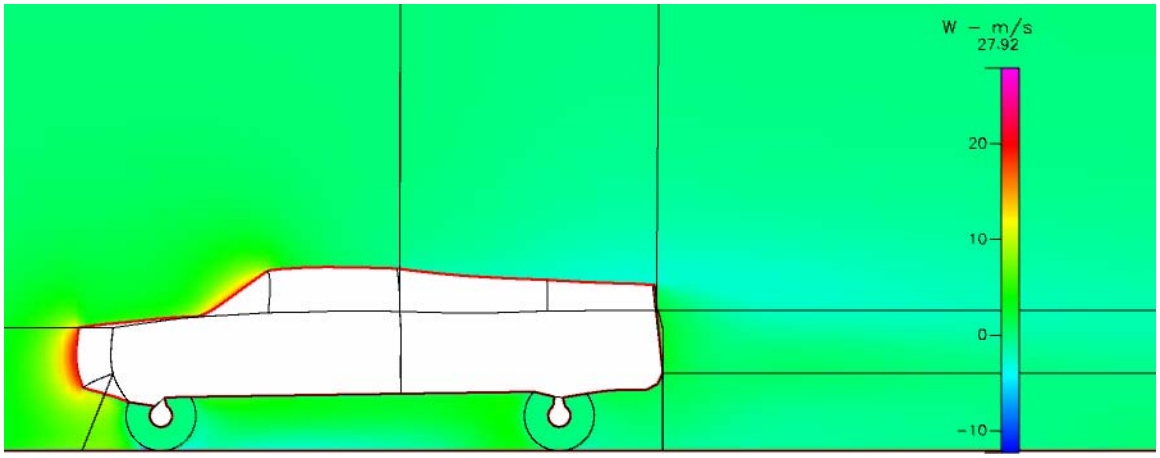

Figure 113. 3D Curved Canopy: Z Velocity Distribution .4 M from Centerline

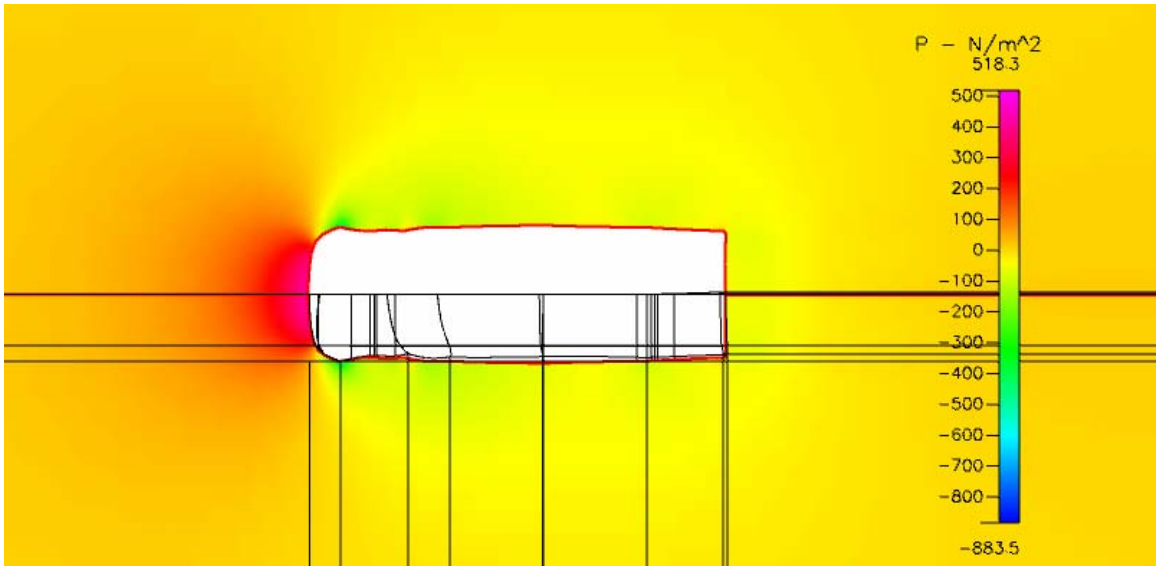

Figure 114. 3D Curved Canopy: Pressure Distribution Top View

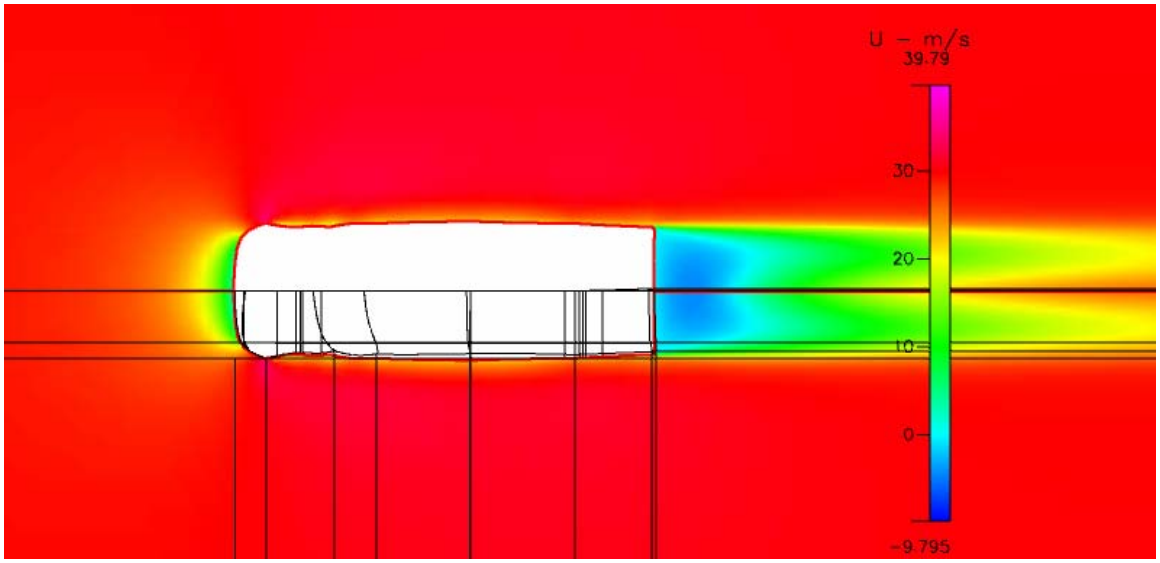

Figure 115. 3D Curved Canopy: X Velocity Distribution Top View

12. 3D Curved Canopy, Air Dam and Side Skirts

In keeping with the previous analyses, a 3D Curved Canopy with the addition of an air dam and side skirts was developed. This will allow for the comparison with multiple combinations of vehicular accessories.

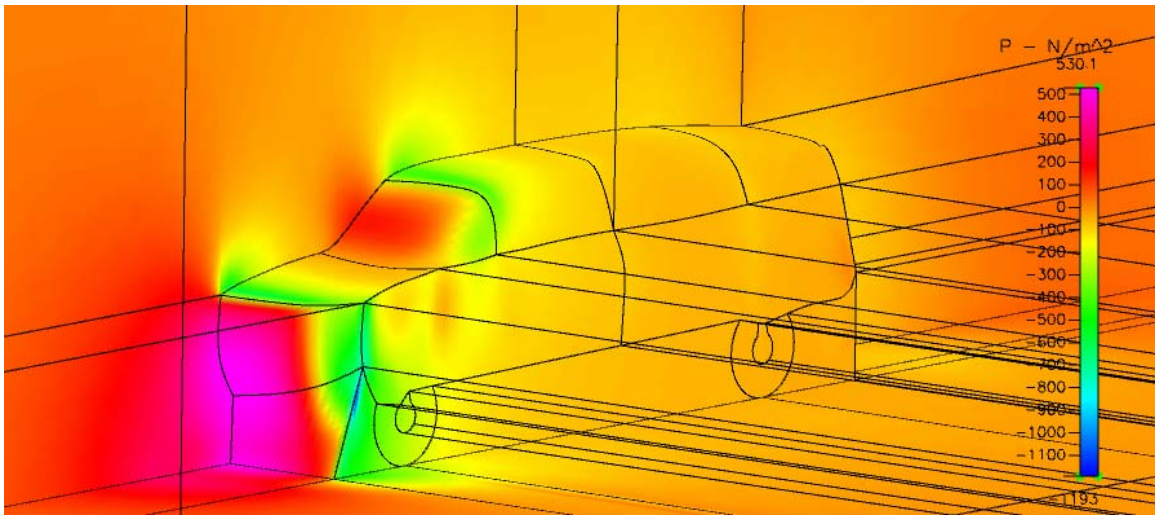

Figure 116. 3D Curved Canopy, Air Dam and Side Skirts: Isometric View of Pressure Distribution

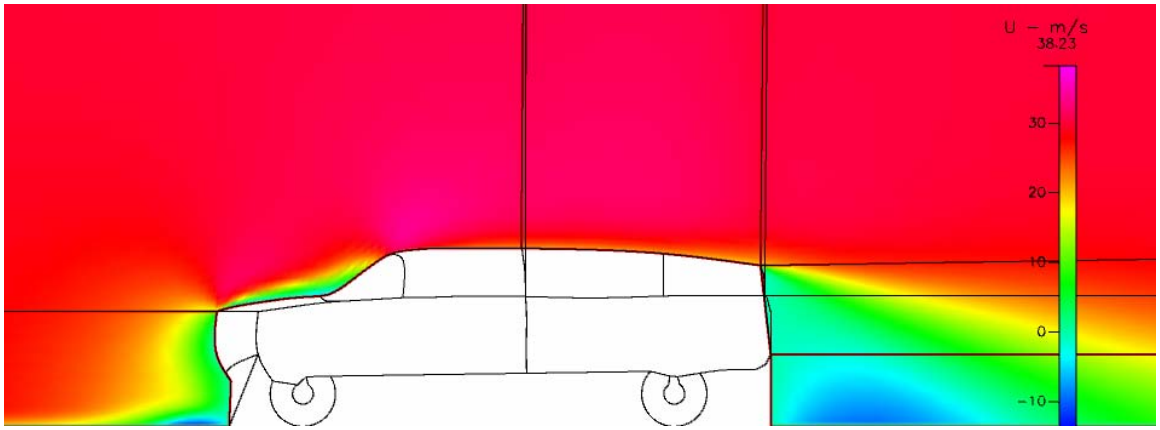

Figure 117. 3D Curved Canopy, Air Dam and Side Skirts: X Velocity Distribution along Centerline

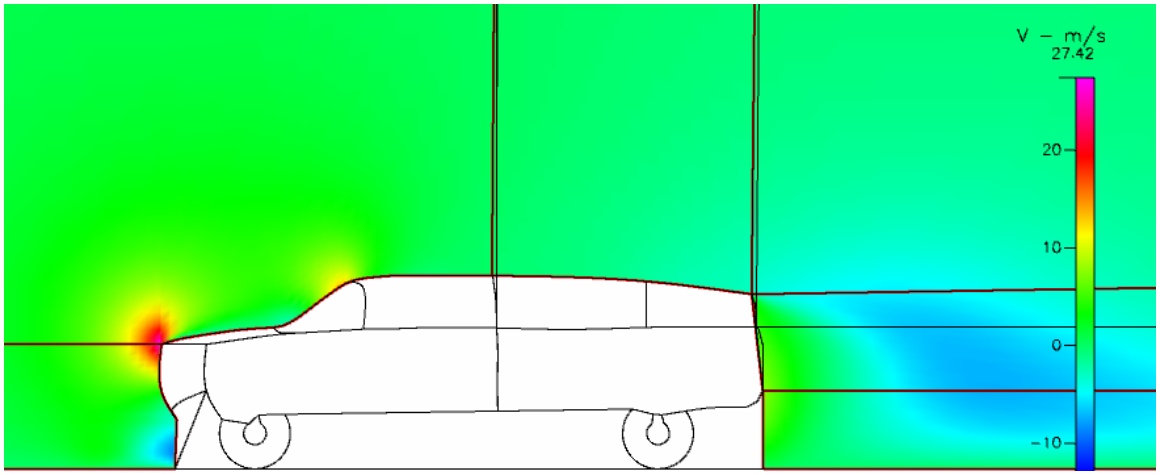

Figure 118. 3D Curved Canopy, Air Dam and Side Skirts: Y Velocity Distribution along Centerline

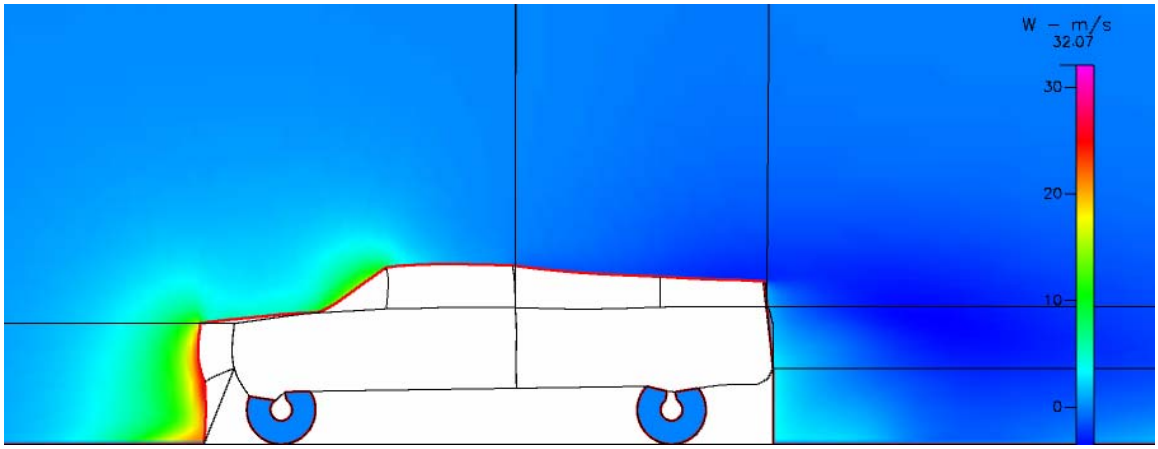

Figure 119. 3D Curved Canopy, Air Dam and Side Skirts: Z Velocity Distribution .4 M from Centerline

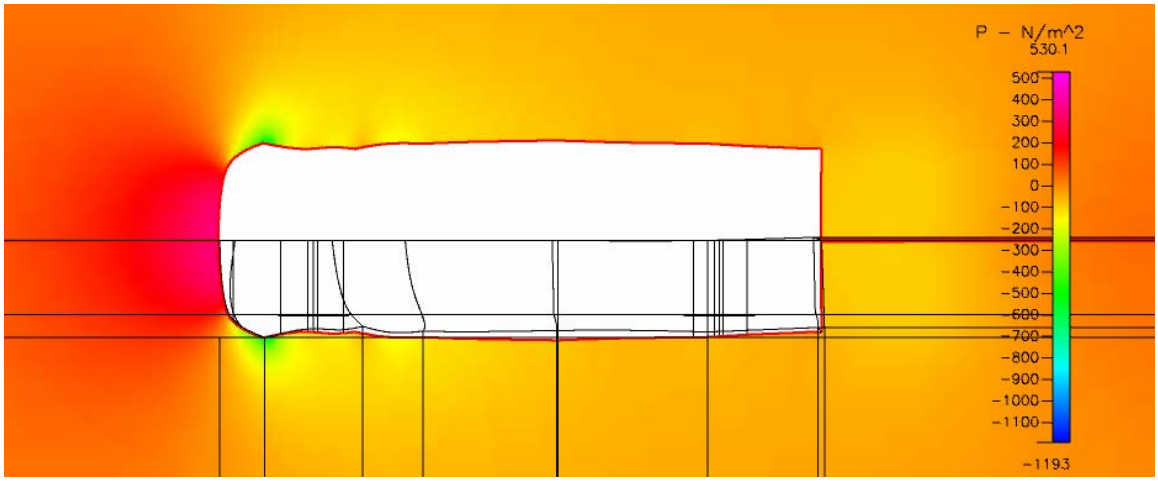

Figure 120. 3D Curved Canopy, Air Dam, and Side Skirts: Pressure Distribution Top View

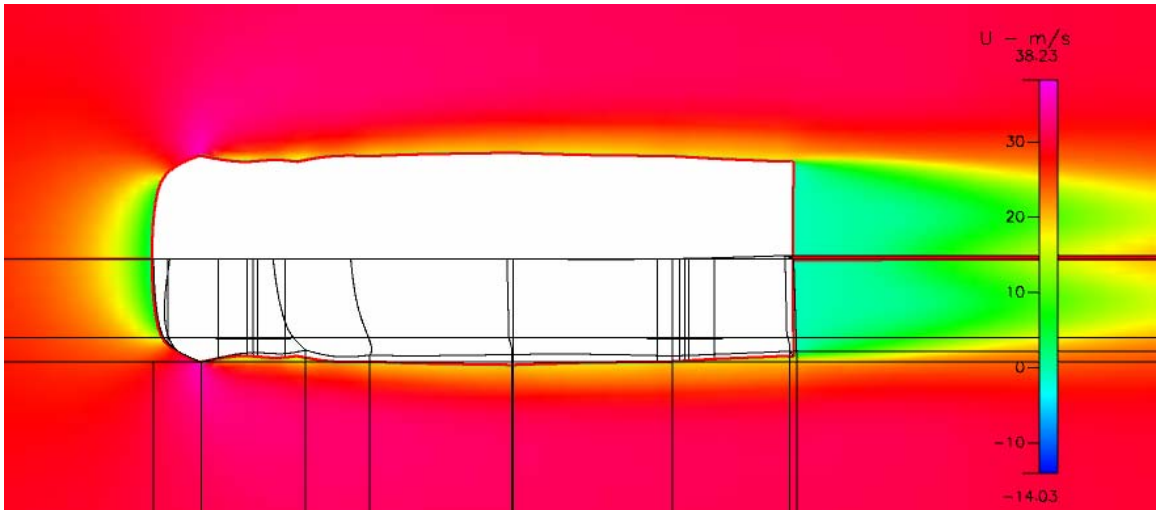

Figure 121. 3D Curved Canopy, Air Dam, And Side Skirts: X Velocity Distribution Top View

13. Underbody Designs

This section was developed to investigate the general effects of underbody design upon drag. The next several figures will show the effects of different air dam configurations, and side skirts. Specifically, it was of interest to the author to quantify the effects of raising the air damn off of the ground by varying heights. In the interest of brevity, only the isometric pressure distribution, x velocity distribution top view, and the pressure distribution top view will be shown in this section. The top views will be slightly different than those from the previous sections because the area under the vehicle is of interest here instead

of the approximate center. As a result, the top view Y plane cuts will be made at the center of the axles at roughly .35 M off of the ground.

a. Baseline

This model uses no accessories to direct the flow under the vehicle and will be used as the baseline for comparison. This is also the overall baseline for the entire project. It is shown below but with emphasis on the undercarriage this time.

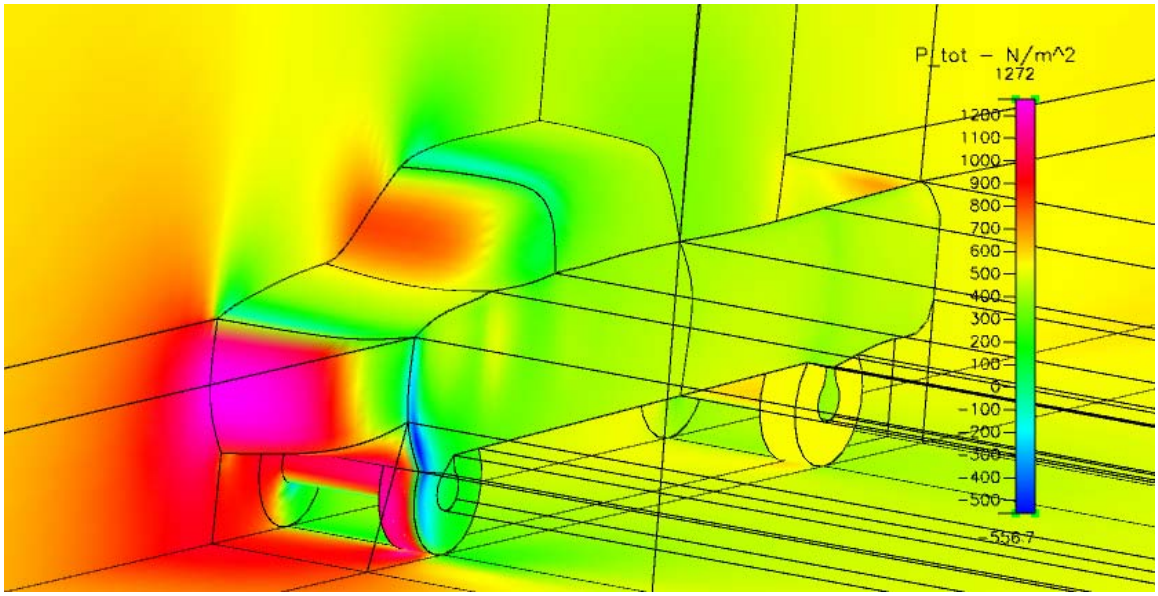

Figure 122. Baseline: Isometric Pressure Distribution

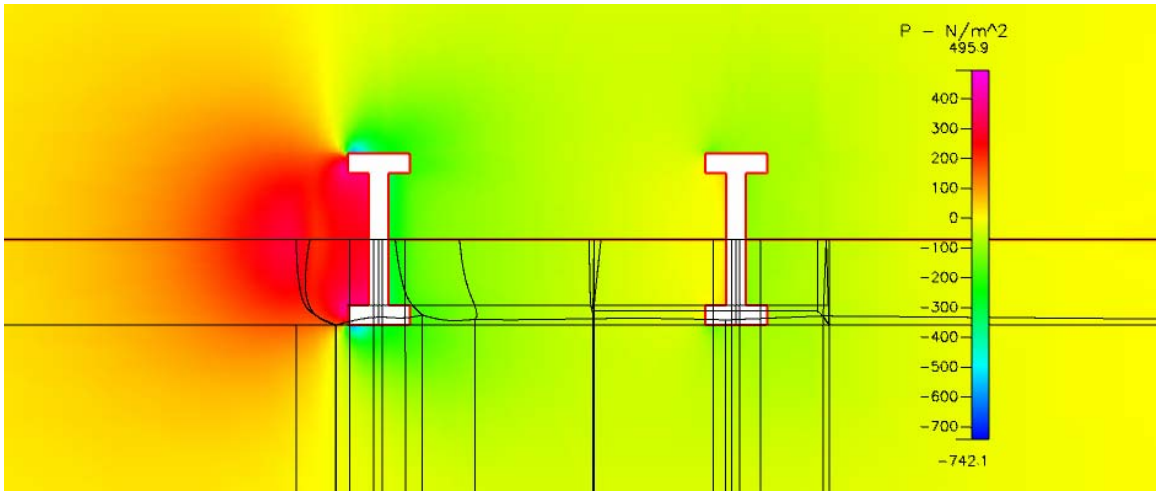

Figure 123. Baseline: Pressure Distribution Top View

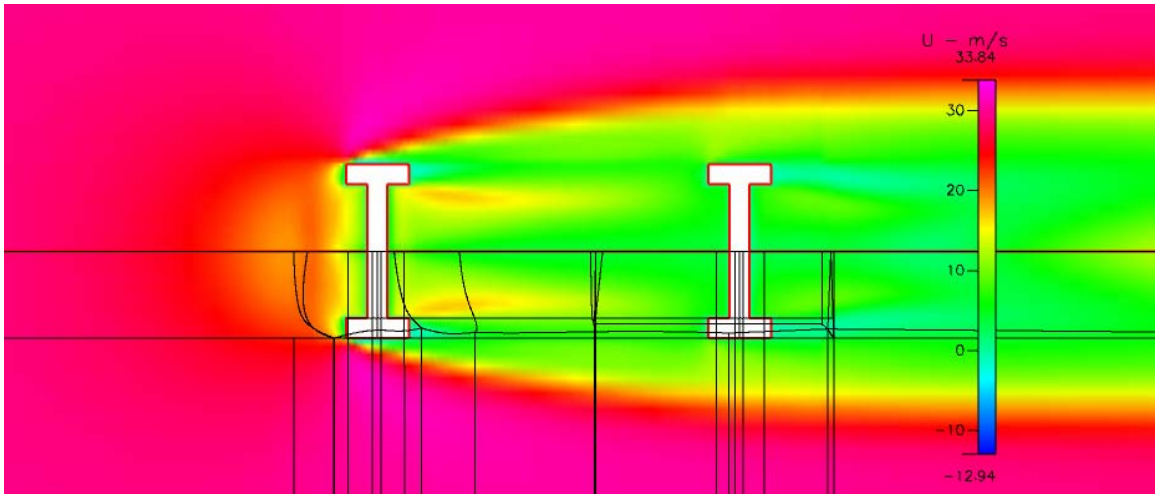

Figure 124. Baseline: X Velocity Distribution Top View

b. Air Dam

This model is the air dam alone with the air dam sealing perfectly against the ground.

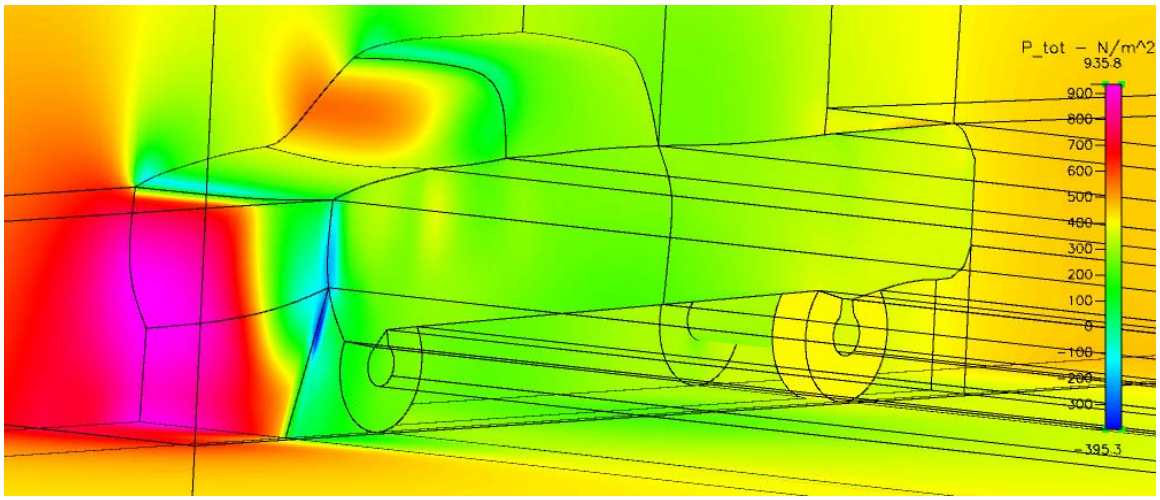

Figure 125. Air Dam: Isometric Pressure Distribution

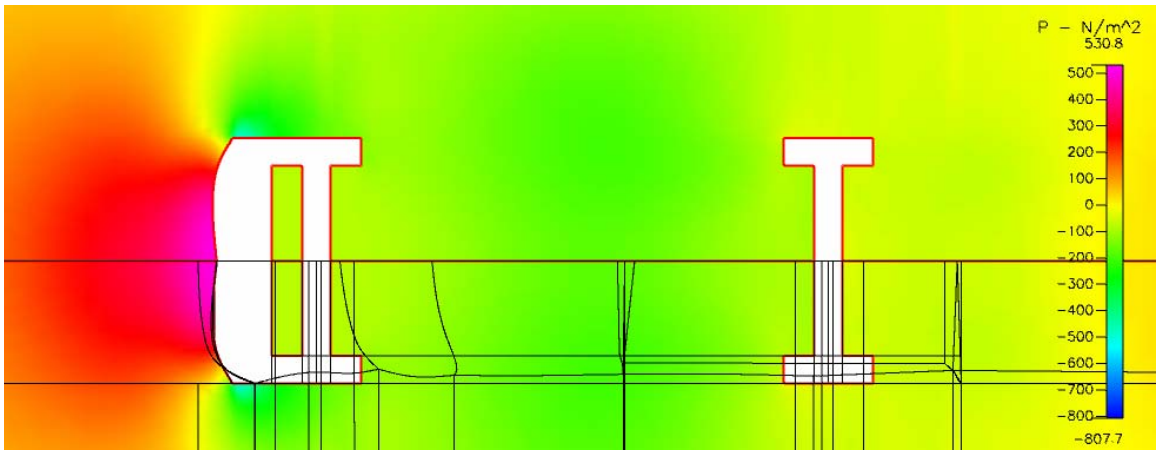

Figure 126. Air Dam: Pressure Distribution Top View

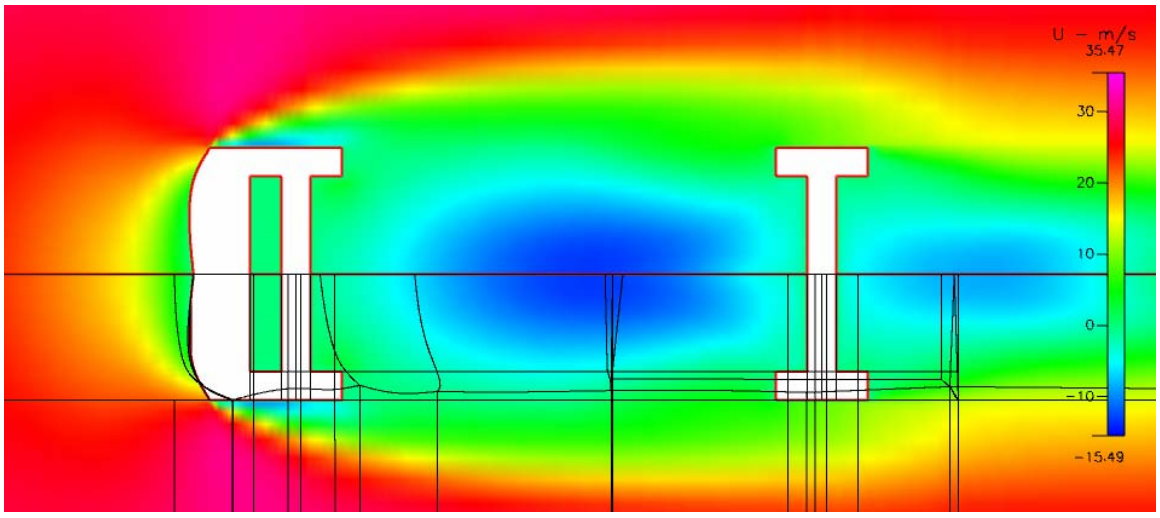

Figure 127. Air Dam: X Velocity Distribution Top View

c. Air Dam: 4 Inches Raised Off of the Ground

The purpose of this design and the next was to see the effects of not having a perfects seal between the air dam and the ground.

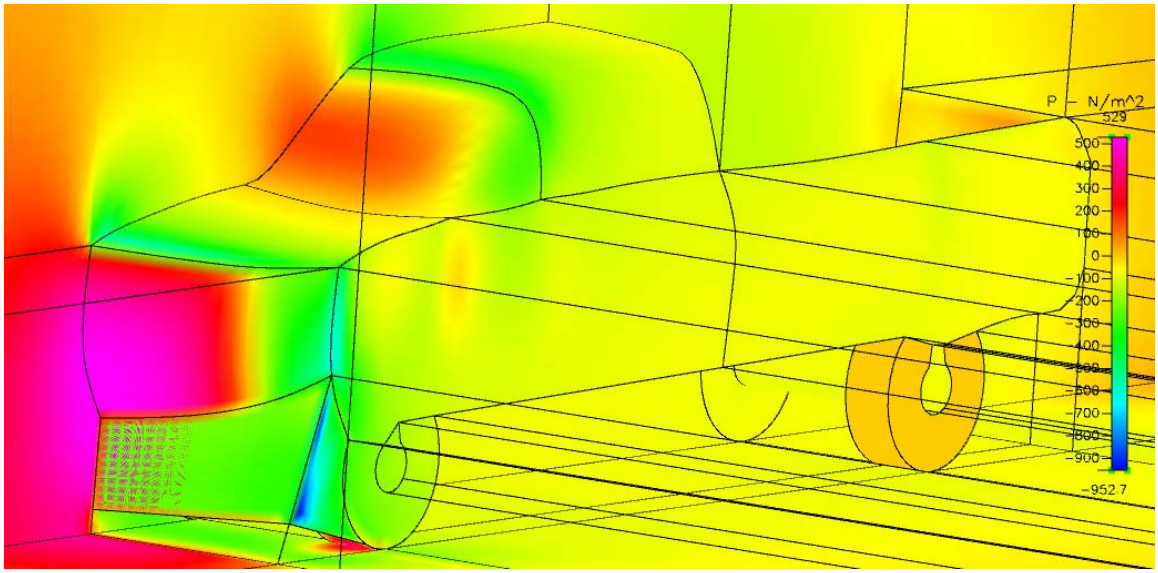

Figure 128. Air Dam with 4 inch Gap: Isometric View of Pressure Distribution

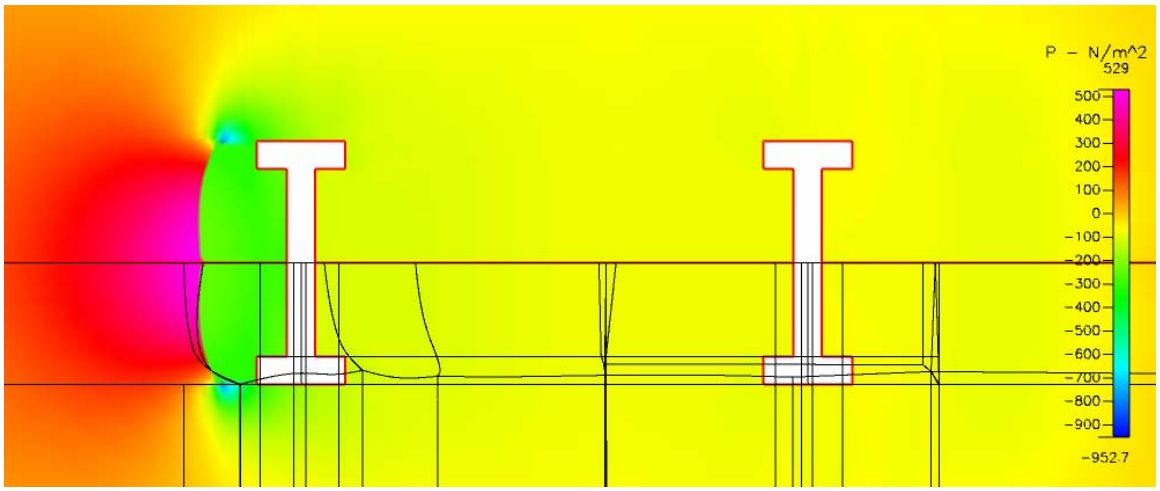

Figure 129. Air Dam with 4 inch Gap: Pressure Distribution Top View

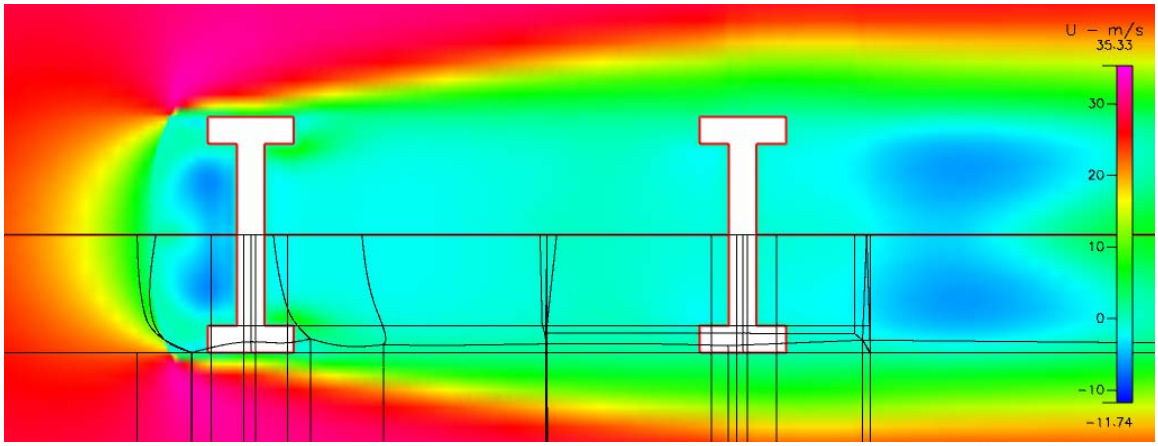

Figure 130. Air Dam with 4 inch Gap: X Velocity Distribution Top View

d. Air Dam: 2 Inches Raised Off of the Ground

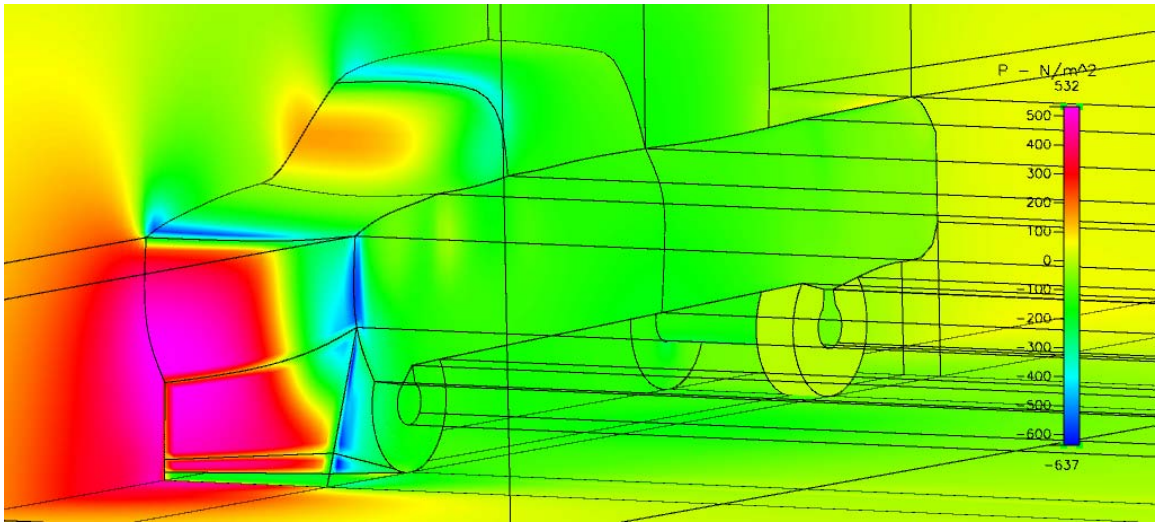

Figure 131. Dam with 2 inch Gap: Isometric View of Pressure Distribution

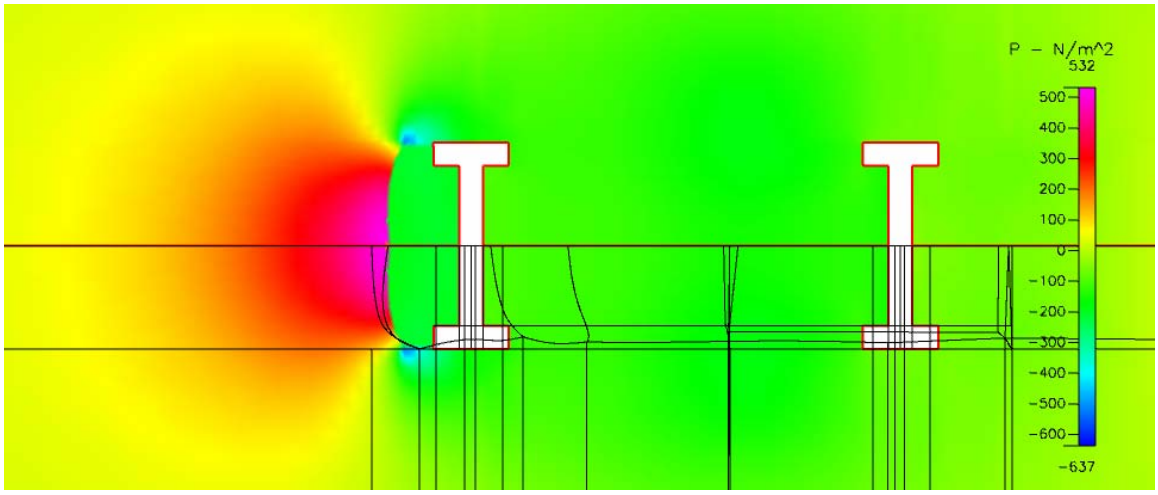

Figure 132. Air Dam with 2 inch Gap: Pressure Distribution Top View

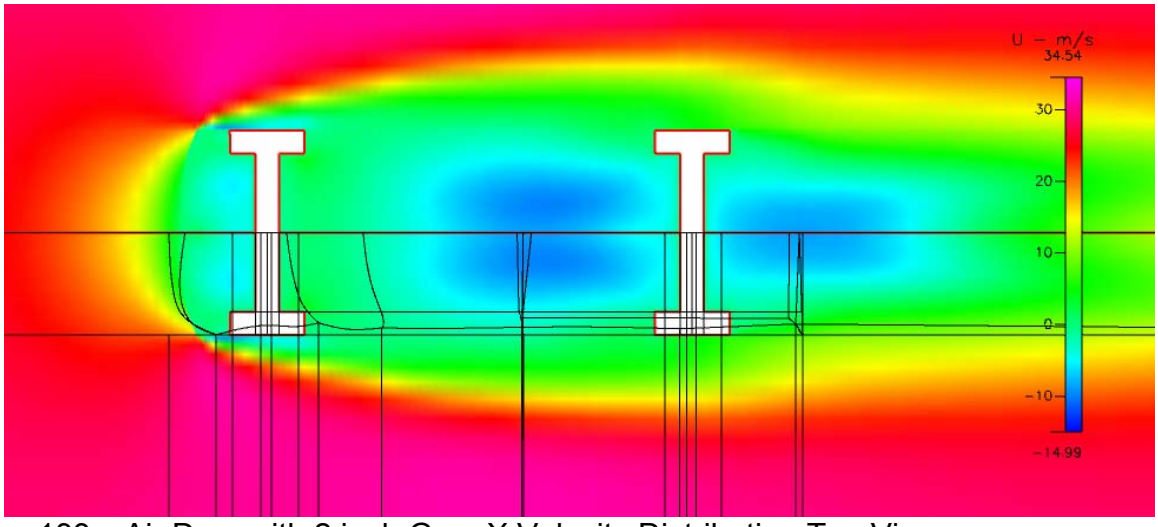

Figure 133. Air Dam with 2 inch Gap: X Velocity Distribution Top View

e. Air Dam and Side Skirts

This model represents the ideal situation where a complete seal exists around the vehicle preventing any intrusion of air.

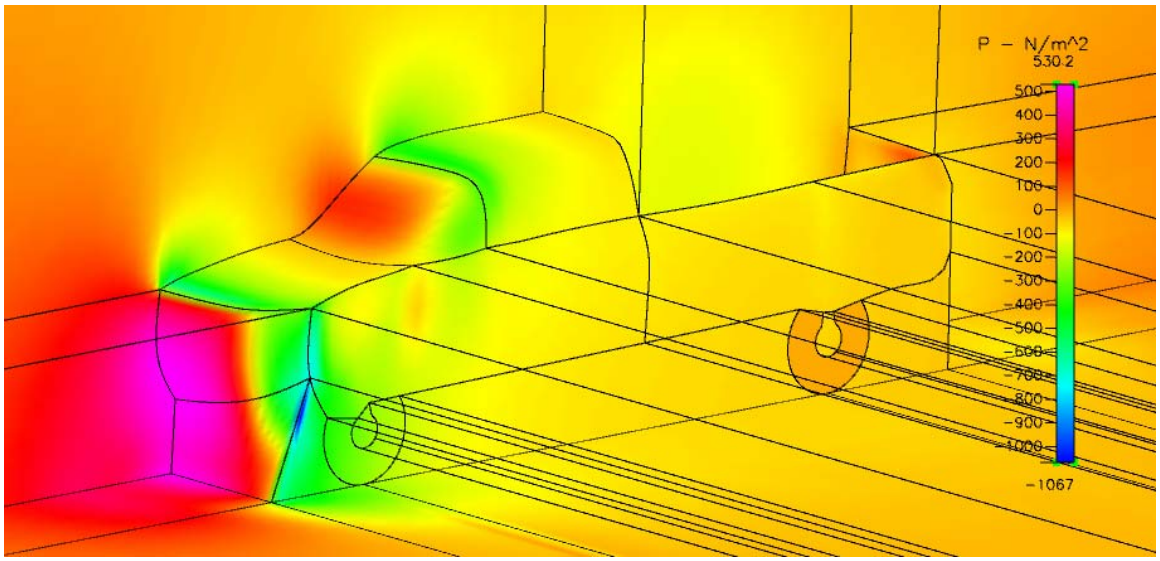

Figure 134. Air Dam and Side Skirts: Isometric Pressure Distribution

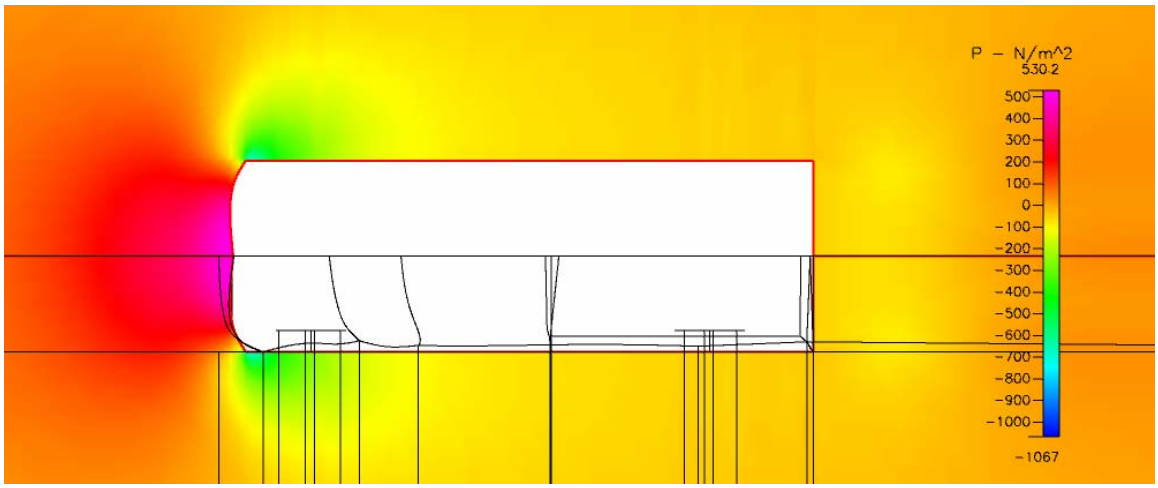

Figure 135. Air Dam and Side Skirts: Top View of Pressure Distribution

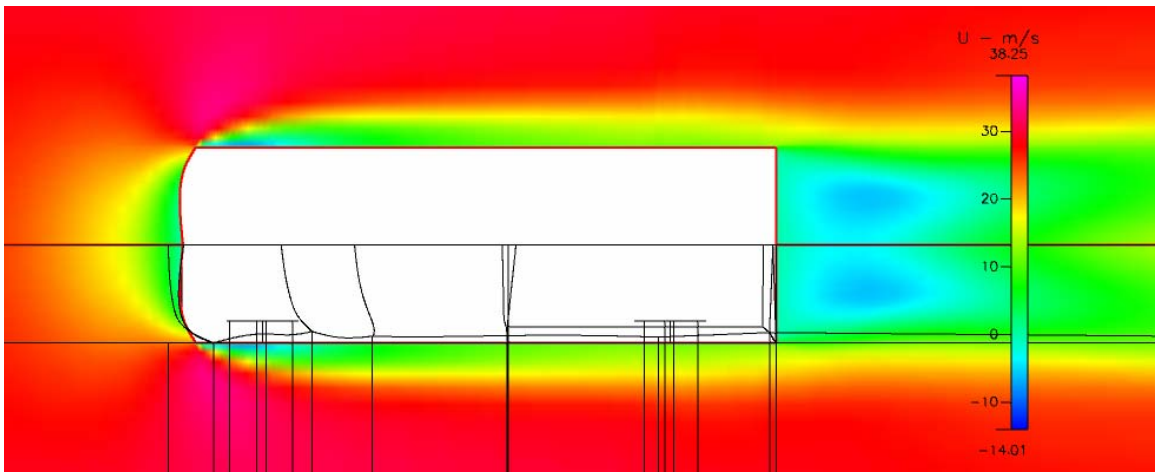

Figure 136. Air Dam and Side Skirts: Top View of X Velocity Distribution

14. Discussion of Three Dimensional Light Truck Shape Studies

While all of the figures in this section are insightful, the most useful information is contained in the force summary from each model. The viscous and pressure forces are summed and tabulated below for each model considered in the preceding sections. As well the coefficient of drag was determined using the following equation and constant values:

$$R_d = \frac{\rho}{2} C_d A_f V^2$$

$$R_d = Drag\ Force\ (N)$$

$$\rho = Density\ of\ Air = 1.23\ kg/m^3$$

$$A_f = Frontal\ Area\ (m^2) = 1.82\ m^2$$

$$V = Velocity = 30\ m/s$$

Model Design Description	Figures	Drag (N)	Cd	% Difference in Drag
Baseline Condition	50-55	568.52	0.563	
AirDam Only	56-61	482.35	0.478	17.86%
Air Dam and Side skirts	62-67	457.36	0.453	24.31%
2-D Optimal Canopy Only	68-73	529.08	0.524	7.45%
Air Dam and 2-D Optimal Canopy	74-79	440.68	0.436	29.01%
Air Dam, 2-D Optimal Canopy and Skirts	80-85	429.91	0.426	32.24%
Tonneau cover only	86-91	567.71	0.562	0.14%
Traditional Canopy	92-97	528.08	0.523	7.66%
Tonneau Cover Air Dam and Side Skirts	98-103	466.71	0.462	21.82%
Traditional Canopy, Air Dam, Skirts	104-109	435.51	0.431	30.54%
3-D Curved Canopy (Optimization baseline guess)	110-115	515.88	0.511	10.20%
3-D Curved Canopy + Air Dam and Sides	116-121	426.11	0.422	33.42%
Best 3-D optimized Canopy	166	410.69	0.407	38.43%

Table 2. Tabulation 3D Shape Studies
*Note 3D optimization taken from the results of the next section

There are dozens of observations that can be made about the various effects of the designs considered. Trends that the author considers the most important are as follows:

1. Bed covers affect the overall drag of the vehicle. In order of increasing drag the canopies are listed below:
    a. Best Optimized 3D Canopy (see next section)
    b. 3-D Curved Canopy
    c. 2-D Optimized Canopy
    d. Traditional Canopy
    e. Tonneau Cover
    f. Baseline (open bed)
2. An air dam helps all bed cover designs to achieve synergistic results. For example, the run that solved the 2-D Optimal Canopy with air dam and side skirts was actually a better drag reduction (32.24%) than if one just added the effects of the three accessories (31.76%).
3. The air dam alone is more important than any other design element. This is readily observable from the table above. When considered alone the air dam (17.86% reduction) is far more important than any other development.
4. Overall designs are unable to develop their true potential without the air dam. The air dam protects the vehicle from developing high pressure regions on the front of both axles and wheels and allows the air to slip around the vehicle. It is clearly evident that high pressure develops on the front of the air dam but it is not as detrimental as if it were on both axles and sets of wheels.
5. Boundary layer separation increases drag but must be evaluated relative to the negative pressure developed normal to the backside of the vehicle. For example, the boundary layer on the traditional canopy does not separate until the very end of the vehicle is reached which is similar to the optimized canopies. The reason the optimized canopies are slightly better in terms of drag is that there is less area at the tail of

the vehicle to interact with the negative pressure that results from the turbulence and boundary layer separation. This is truly where the optimizer becomes critical. Essentially, the objective function is compromising between boundary layer separation and the area of the design that is exposed to the negative pressure.

6. The air dam works best when it extends all the way to the ground. While this may appear to be impractical for trucks it is possible that the air dam could accomplish this by being made from flexible materials or having a sensor that raises and lowers it when an impact is imminent on non-highway situations. An in-cab control is also a possibility. The table below summarizes the negative effects of having a raised air dam, no air dam, and the effects of side skirts.

Model Design Description	Figures	Drag (N)	Cd	% Difference in Drag
Baseline Condition	122-124	568.52	0.563	
AirDam Only	125-127	482.35	0.478	17.86%
Air Dam with 4" Gap	128-130	504.35	0.500	12.76%
Air Dam with 2" Gap	131-133	487.09	0.482	16.72%
Air Dam and Side skirts	134-136	457.35	0.453	24.31%

Table 3. Tabulation of Underbody Related 3D Shape Studies

D. SHAPE OPTIMIZATION

The problem formulation section of this thesis describes in general terms the composition of a modern shape optimization using CFD. The section is dedicated to showing how a shape optimization can be performed using a commercially available program. Subsection one details the method used by the author to perform an optimization in CFD-ACE. The subsequent subsections discuss the results of the two dimensional and three dimensional optimizations respectively.

1. Methodology Used to Perform Shape Optimization in CFD-ACE

There are several ways to perform parametric studies and optimizations using CFD-ACE. Most of these methods are geared toward the simplistic

optimization of a single characteristic of a single boundary condition. These methods are nearly automated and require very little subsurface knowledge of the analyst. Unfortunately, these methods are not adept at solving complicated problems involving multiple characteristics of several boundary conditions. This was precisely the interest of the author.

To perform this thesis it was necessary to optimize the summation of all of the viscous and pressure forces on a multitude of boundary conditions. To do this, the analyst must be able to develop stable solutions to a baseline CFD problem, write pseudo-code in a proprietary language, develop grids using an open source language (PYTHON), and program an objective function using PYTHON. All of these supporting programs will be accessed by CFD-ACE during the optimization. Careful attention must be paid during the setup of an optimization because, in many instances, the first function call will require several hours to compute and is the only source of feedback. The reader should refer to the UML model proposed in figure 9 throughout this section for clarity. It is assumed that the reader has some basic knowledge of CFD as terms specific to CFD will be used throughout this section.

a. Step One – Develop a Stable Baseline Geometry and Solution

The first step in doing a shape optimization should involve creating a stable mesh that is grid independent and converges easily whenever possible. In CFD-ACE suite one should create the grid in CFD-GEOM and then make sure that it reaches a converged solution (residuals dropping by 4 orders of magnitude) in CFD-GUI. The process of developing a stable and well conditioned geometry in CFD-ACE is detailed in their user manuals [Ref. 12]. It is worth noting thought that obtaining a stable solution often requires trial and error and significant experience.

b. Step Two – Develop a Variable Geometry Grid

This requires some knowledge of the open source language PYTHON. Almost all commands required to develop a shape optimization are

contained in the ensuing code. If the reader requires greater knowledge of PYTHON the author would like to suggest that they start at www.python.org.

CFD-GEOM is the preprocessor or grid generator for CFD-ACE. CFD-GEOM can be controlled with PYTHON commands. The first step in generating a variable geometry/mesh is to open the stable geometry that was created in Step One using CFD-GEOM. Once the stable model is open, all volumes that come in contact with the faces that are to be shape optimized should be deleted. Then the faces themselves should be deleted, followed by the edges and points that make up these faces. These will be reconstructed later using variables. The author found it useful to write down the coordinates of the points controlling the edges so that the reconstruction would go smoothly. Now that the stable geometry is stripped down to the essential components that won't be optimized the grid should be saved as a CFD-GEOM .ggd file. It is of the utmost importance that the analyst be very careful not to alter the baseline file from this point forward.

As well, two settings in CFD-GEOM need to be checked at this point. First, the analyst should make sure that journaling is turned on. In the 2002 version of CFD-GEOM this setting is found under settings->Preferences->Tools->Journaling. The other setting that has to be checked is also in the preferences window. The analyst must ensure that the PYTHON scripts directory is pointed to 'c:\Program Files\CFDRC\V2002\ScriptLanguage'.

Next, the deleted geometry will be reconstructed using design variable that will later be accessed by the optimizer. The author's preferred method was to open a PYTHON editing window from within CFD-GEOM. See the figure below:

Figure 137. Accessing the PYTHON Script Editor

This launches a PYTHON editor that will be used to develop the variable geometry file. Comment lines start with the # symbol and will be used to explain what is happening in the code. The first lines of code should be as follows:

```
1.  #Variable Geometry Scripting File
2.  # By Nathan Williams
3.  #     LT, CEC, USNR
4.  #    Aug 13 2003
5.  # This file was generated by CFD-GEOM
6.  # Lines ten through twenty one import subroutines from the
7.  # CFD-GEOM library.  These routines will be necessary for creating
8.  # the variable geometries and for importing the baseline file.
9.  #
10. import GPoint
11. import GCurve
12. import GSurface
13. import GLoop
14. import GEdge
15. import GFace
16. import GBlock2D
17. import GBlock
18. import GEntity
19. import GManip
20. import GUnstruct
21. import GInterface
22. #
23. #
7.  # Set the geometric precision of the model
8.  # Line 23 sets the geometric precision using the subroutine GEntity
9.  GEntity.SetPrecision ( 1E-006 )
10. #
11. # The Subroutine GInterface imports the baseline model
12. GInterface.GGDRead('C:/Scripts/Baseline.GGD')
13. #
14. # Next the variables used to specify the geometry are defined
15. # Names relevant to the body of the shape should be used
16. #The Subroutine GEntity is used to define the variables
17. #The initial values for the design variables are specified here.
18. #
19. #
20. backmove = -.250489
21. GEntity.CreateParameter (globals (), 'backmove')
22. midedgey = 1.64608
23. GEntity.CreateParameter (globals (), 'midedgey')
24. midcentery = 1.74108
```

25. GEntity.CreateParameter (globals (), 'midcentery')
26. backedge = 1.57001
27. GEntity.CreateParameter (globals (), 'backedge')

At this point the analyst should run the script to ensure that the grid script works and to start using CFD-GEOM to code the rest of the file through the graphical user interface. This can be accomplished because the feature journaling is on. In lines 37 through 44 above, four design variables were specified. This was the number of design variables used by the author for the three dimensional optimizations. The complexity of the shape to be optimized and the surrounding grid will dictate the number of design variables necessary to recreate the baseline grid. These variables are now used inside the CFD-GEOM graphical user interface to place the control points for the geometric optimization. A control point controls the curves, edges, faces and therefore volumes that make up the overall grid.

To place a control point the analyst should select the button in CFD-GEOM shown below:

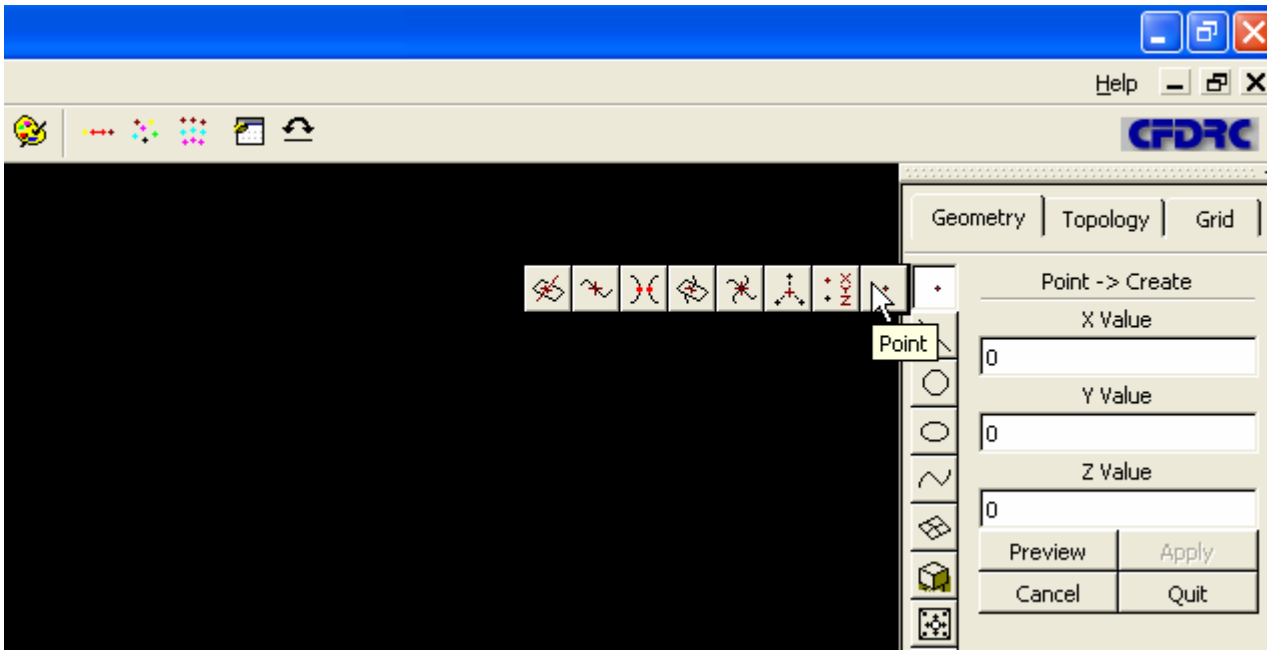

Figure 138. Placing a Control Point

After selecting the point tool shown above the analyst should put the variable of interest into the X Y or Z Value field shown above. In the case of this thesis, for example, the author systematically placed the variables backedge, backmove, midedgey and midcentery in Y Value field as the points were placed. Obviously, three variables could be used to specify each control point. In the interest of time only Y values were used to in this thesis. As the points are placed by pressing apply, CFD-GEOM will automatically code the following lines:

```
24. #Control points are place using CFD-GEOM
25. # the subrouting GPoint.Create specifies the X, Y and Z values
26. #
27. geom_point6208 = GPoint.Create (-17.5, midedgey, .8)
```

Lines similar to number 48 were included by CFD-GEOM in the PYTHON script to place points for the other three variables. From here the analyst should attempt recreate the baseline grid from the control points. The author took care to place the control points in the exact position that was used for the original points in the baseline model. The analyst will have to create curves or lines through the control points. The lines are followed by edges, faces and volumes that make up the grid. As each of this constructs are created more code will be added by CFD-GEOM to the user's script file. In actuality, the analyst will very likely end up with pages of code that they don't understand and contains many lines that are superfluous as mistakes are made and erased. All of these actions will be recorded in the code. This is not a problem and the author chose to leave the superfluous code in the script as all attempts to delete it generated errors on subsequent runs.

The last step in creating a usable variable geometry script is to save the file in CFD-GEOM as a .dtf. Regular users of CFD-GEOM will be intimately familiar with this step as it is necessary to create a grid that is acceptable for CFD-ACE. After this is done a new line of code will appear in the script file and the script file is complete. The line is shown below:

```
28. #Now save the model as a DTF file and the scripting is complete
29. #GInterface is used to write the DTF file
30. #
```

31. GInterface.DTFWrite_3d ('3DOpttimal.DTF')

One of the author's entire scripts is shown in Appendix A. Obviously this code is not easy to read as it was generated automatically so it contains some superfluous code. The analyst should selectively change the design variables in the code and rerun the script after each change. If the variables work to develop the geometries of interest, the next step should be commenced.

c. Step Three – Write Psuedo Code to Assemble Objective Function

This step is far easier than the previous one but requires some knowledge of boundary integral pseudo code method used by CFD-ACE. The DTF file created at the end of the last step should be opened at this point. All settings CFD flow settings can now be set up. For instance, one can now set all boundary conditions, initial conditions, volume conditions, and solver conditions. The reader is referred to the CFD-ACE user manual for further instruction.

Within CFD-ACE/CFD-GUI it is the most convenient to name the boundaries of interest to the analyst. This can be accomplished by selecting the BC (boundary condition) tab. All boundaries are available in the lower left hand window. By right clicking on an individual boundary it is possible to select properties and specify the name of the boundary or face. See figure below:

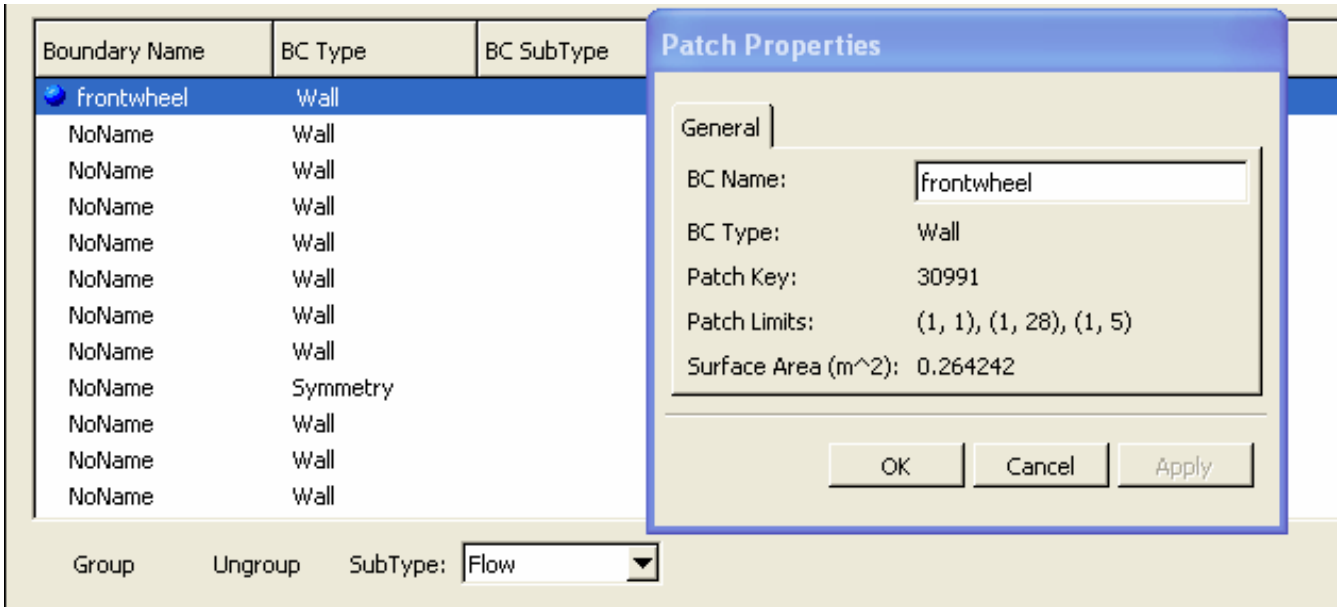

Figure 139. How to Name a Boundary Condition

In this example the front wheel boundary condition is named. This should be done for all faces that will be accessed by the cost function. In order for the cost function to gain access to all of the boundary conditions of interest is necessary to write a boundary integral file with the extension .fmt. This file should have the same name as the DTF file and reside in the same directory.

The language used is essentially a pseudo code. The boundary integral file used for this thesis is shown below:

```
Surface  : cabtop
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface  : canopy1
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface  : canopy2
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface  : back1
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir
```

Surface : back2
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : back3
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : rearquarter
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : rearwheel
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : rearhub
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : rearside
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : skirt
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : frontside
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : frontwheel
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : fronthub
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : damside
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface : damfront

```
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface  : nose
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface  : hood
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Surface  : windshield
Variable : Pressure Force in X_dir
Variable : Shear Force in X_dir

Assemble 1 : cabtop, canopy1, canopy2, &
             back1, back2, back3, &
             rearquarter, rearwheel, rearhub, &
             rearside, skirt, frontside, &
             frontwheel, fronthub, damside, &
             damfront, nose, hood, windshield
```

The pseudo code is fairly simple to discern. The first line of each block contains the boundary condition of interest. The subsequent lines should specify the properties of the boundary that will be used in the cost function. This thesis used the pressure and viscous forces to optimize the light truck shape. As such, the pressure and shear properties were specified after each surface. The reader should go to the CFD-ACE user manual to find the other properties available (there are many) if they are of interest. The syntax must be followed explicitly as shown above.

After the surfaces/faces/boundaries are accessed it is necessary to write another block of code that adds all of the forces. The “Assemble 1” command and subsequent code adds all of the forces and places the resultant in a file for future access by the cost function. The analyst should also be aware that this boundary integral file must have the same name as the .dtf file developed in step two. In order for CFD-ACE to produce to use the boundary integral file a setting change must be made in CFD-GUI. Under the tab ‘out’ and

sub-tab ‘print’ the user should check ‘BC Integral Output’ as shown below:

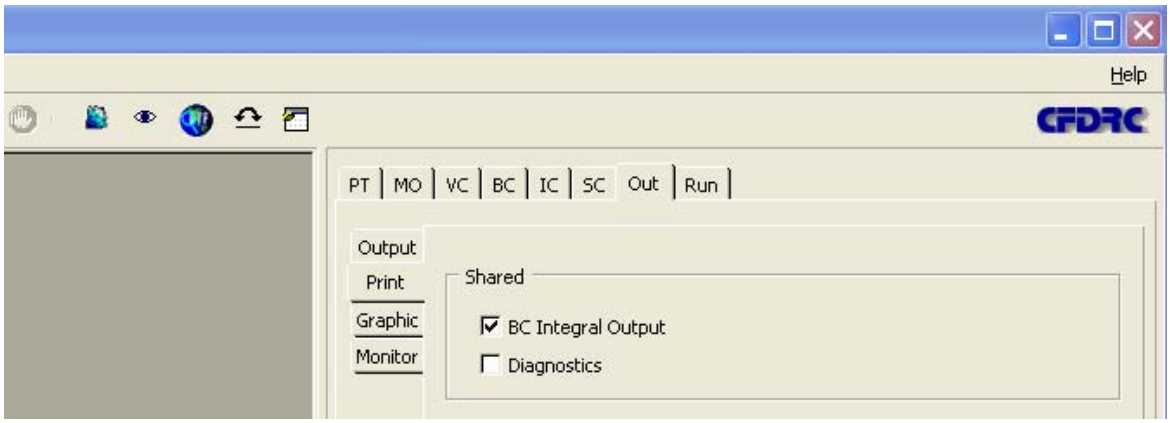

Figure 140. Selecting BC-Integral Output

As a result, the file produced by the assemble command written in the .fmt file will be something like 3DOpttimal.00001_ASSEM_01.dat if we are to continue with the example from the previous steps. The assembled file will be a text file that will be read by the objective function. It is a good idea to do a full trial run in CFD-ACE at this point to ensure that boundary integral file is working. If it is working properly an assembly file should show up in the same directory as the DTF file. The assembly file will appear as a text file as shown below:

# Case	Iter	Pressure Force in X_dir	Shear Force in X_dir
0	291	3.982224E+02	1.246208E+01

d. Step Four – Write an Objective Function in PYTHON

Step four was the most difficult step for the author in terms of programming because it was very difficult to understand exactly how CFD-ACE would use the objective function. There are minimal references in the CFDRC manuals as to how to properly frame an objective function or cost function so that the CFD-ACE optimizer can interact with it. As mentioned before, this is the most flexible method of optimization available in CFD-ACE because it allows the user to access multiple properties of multiple boundary conditions.

In the case of this thesis, we were interested in the resultant of the viscous and pressure forces which act on a dozen surfaces. Writing a user script or objective function in PYTHON is the only way to accomplish this. It is recommended that the reader and future users simply modify the objective

function contained in this section for their particular needs. This objective function can be easily modified to optimize any property of interest. Once again the PYTHON code will be annotated with lines starting with the # symbol:

```
1.  # Cost Function
2.  # Nathan Williams
3.  #LT, CEC, USNR
4.  #15 Aug 2003
5.  #
6.  #start by importing the string function so that the assembly file can be
7.  #read
8.  #
9.  import string
10.#
11.#here a function is defined to compute the cost function for each DTF
12.#file in the optimization.  It is necessary to do this because each
13.#optimization run develops a sequentially advancing DTF file name.
14.def CostFunction(dtf_filename, cost_value):
15.
16.    def RemoveDTF (str):
17.        if (string.find (str, '.DTF') != -1):
18.            str = string.replace (str, ".DTF","")
19.
20.        if (string.find (str, '.dtf') != -1):
21.            str = string.replace (str, ".dtf","")
22.
23.        return str
24.#The lines above essentially takes the DTF filename given to
25.# the cost function by CFD-ACE and strips the DTF extension
26.#off.
27.#
28.#Now the function is executed to remove the DTF extension
29.    dtf_filename = RemoveDTF(dtf_filename)
30.#
31.#The stripped down string now has the assembled boundary
32.#integral output extension added to it and named dat_filename
33.    dat_filename = dtf_filename+"_ASSEM"+"_01.dat";
34.#Now the cost function is able to open the boundary integral.
35.# a variable f will contain the entire text file
36.f=open(dat_filename,'r');
37.#
38.#84 characters are read
39.    f.read(84)
40.#
41.#13 characters are read from the text file and saved as the
42.#Pressure force in the X direction
```

```
43.    PX=f.read(13);
44.#
45.#14 characters are read from the text file and saved as the
46.#shear force in the x direction
47.    SX=f.read(14)
48.#
49.#the shear and pressure and now turned into floating point numbers so
50.#that they can be added.
51.    PXnum=float(PX);
52.    SXnum=float(SX);
53.#
54.#The forces are now added.  This is the primary reason that the user
55.#script method was employed in the first place.
56.    cost_value=PXnum+SXnum
57.#
58.#The drag on the vehicle is returned to the optimizer to aid in selecting
59.# the next search direction.
60.    return cost_value
```

At this point the analyst should open a PYTHON Idle window and run the cost function to ensure that it returns the value expected from their boundary integral output. PYTHON's editor and debugger are known as Idle and are included in the CFDRC distribution.

e. Step Five – Setup the CFD Run in CFD-GUI

This section shows how to setup up an optimization run in CFD-GUI. First the analyst should open the DTF file created by the variable script. Dozens of settings will have to be made before the optimization can be run. On the PT tab at the right side of the screen the following should be selected for the problem that has been discussed thus far:

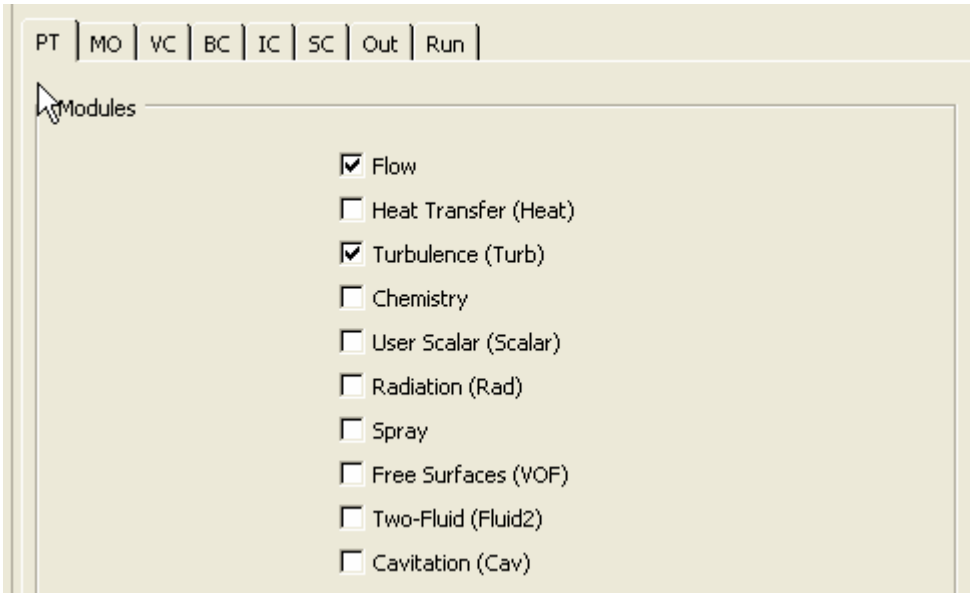

Figure 141. Select the Modules

All of the Volume conditions should reflect the properties of Air:

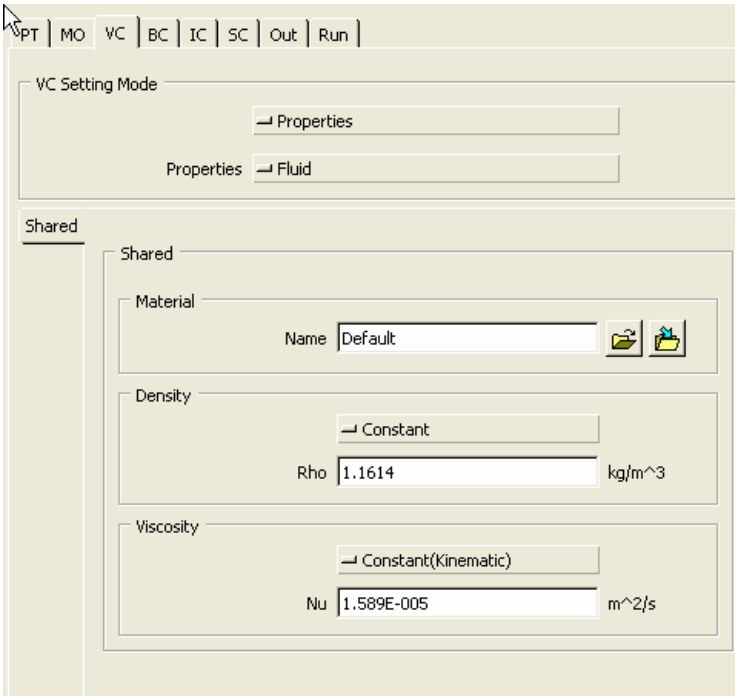

Figure 142. Set the Volume Conditions

The boundary conditions are a critical step. The reader is referred to the CFD-ACE user manual [Ref, 13] to learn how to efficiently set the boundary conditions. Below is an example of how the inlet boundary condition was set:

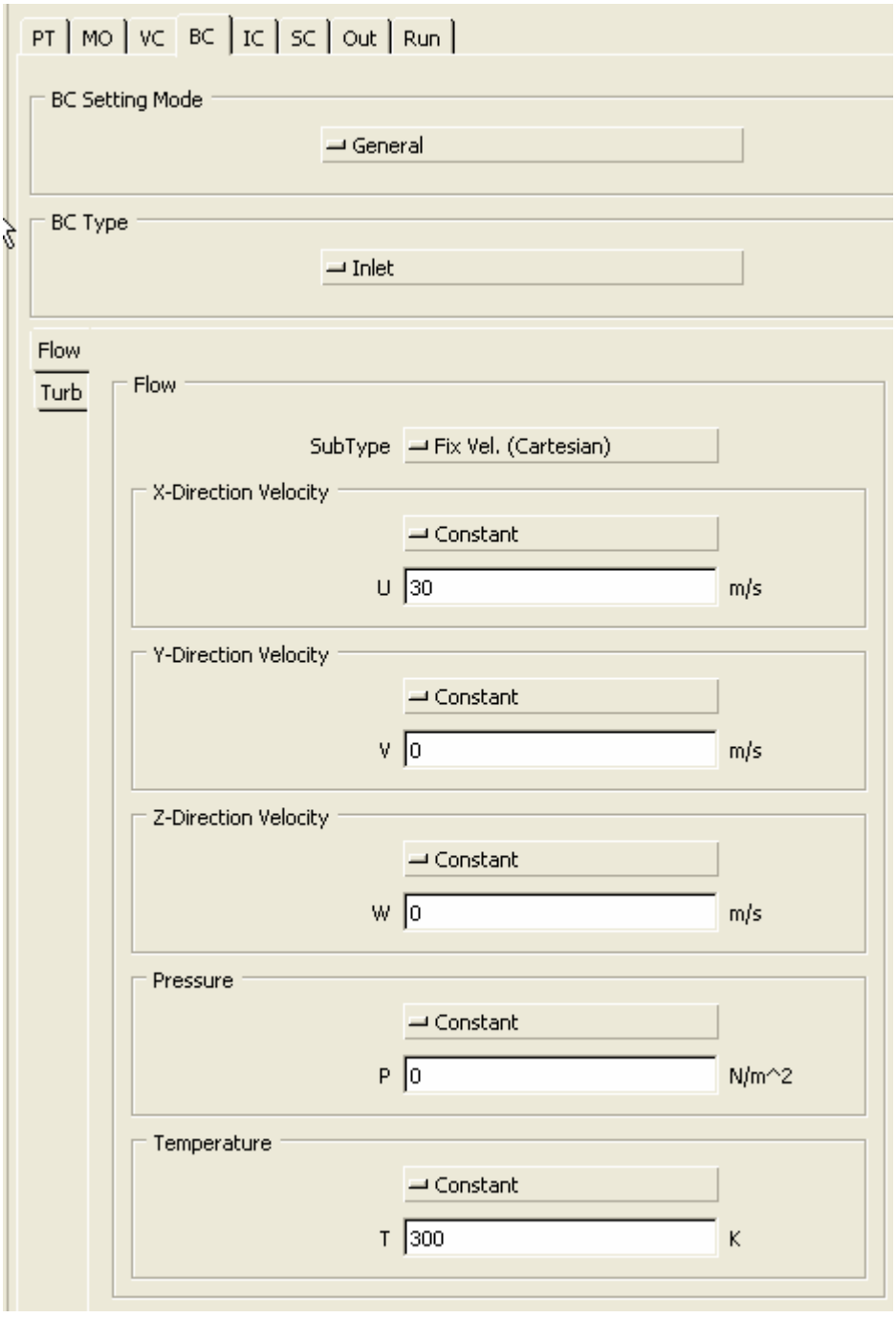

Figure 143. Example of Setting an Inlet Boundary Condtion

Initial conditions are set to reflect the initial state of the system. Since this is not a transient analysis, the initial conditions are of little importance.

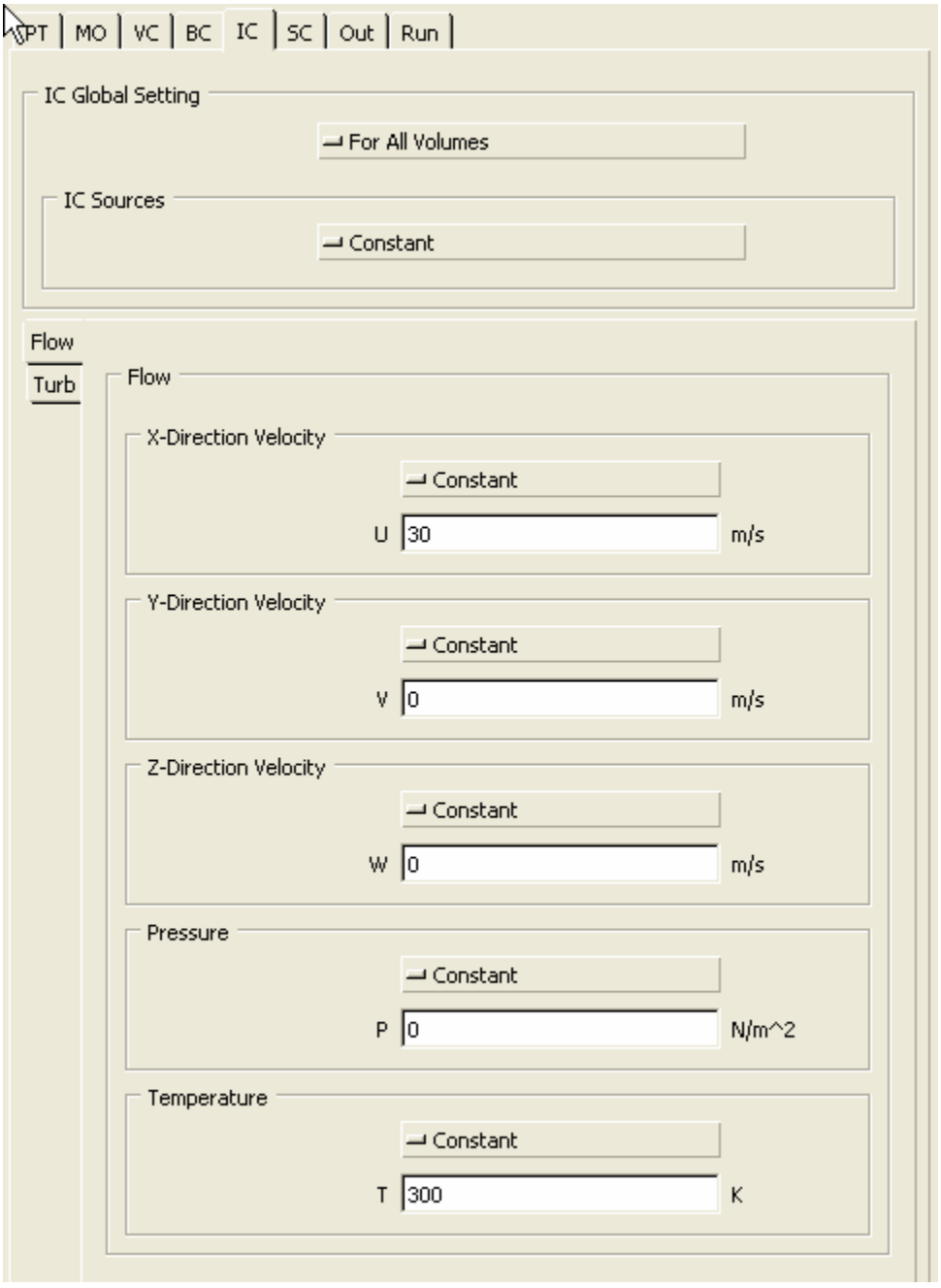

Figure 144. Setting Initial Conditions

The solver tab has several subtabs that require attention for the problem under consideration. First the number of iterations was set to 1000 to ensure convergence. The default convergence criteria were used.

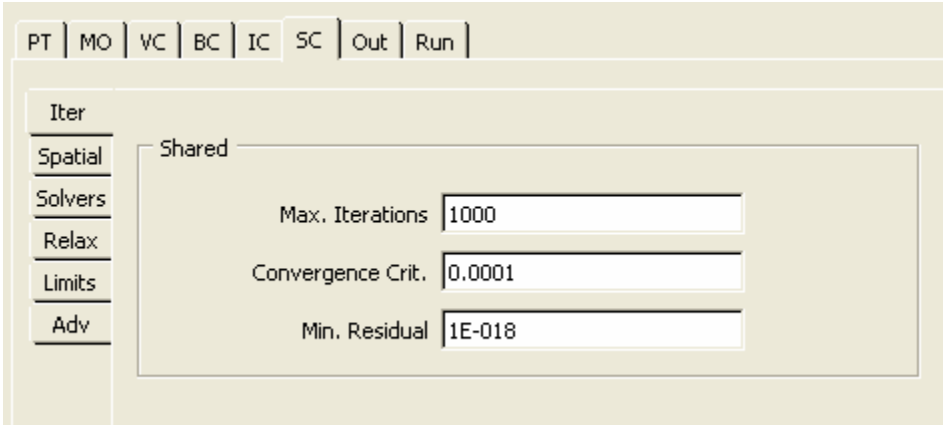

Figure 145. Setting Convergence Criteria

The preferred solver for this problem was found to be the AMG solver after dozens of different iterations:

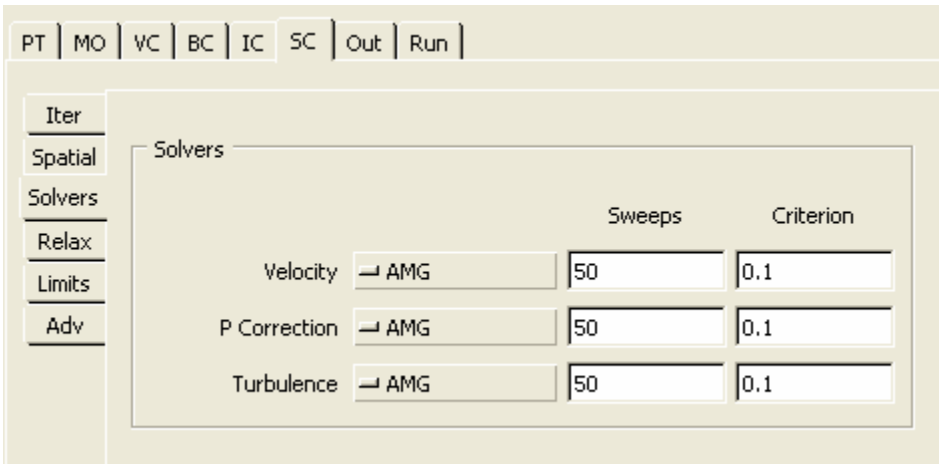

Figure 146. Solver Selection

This solver had to be somewhat relaxed in order to avoid excessive instability in the residuals. These settings were found to provide the

best compromise between stability and convergence time for this problem. CFDRC maintains that these settings can only be adjusted based upon user experience. The following were the setting used:

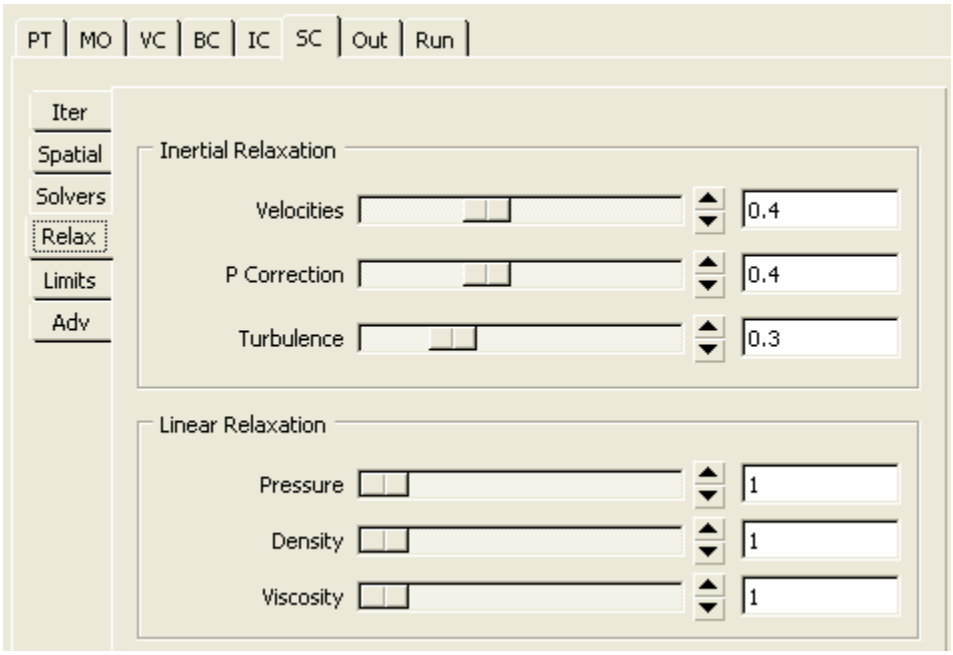

Figure 147. Relaxation Parameters

Under the output tab only one selection is necessary and was mentioned earlier in the chapter. The BC Integral Ouput is critical and the Mass Flow Summary and Force Summary are often convenient for debugging:

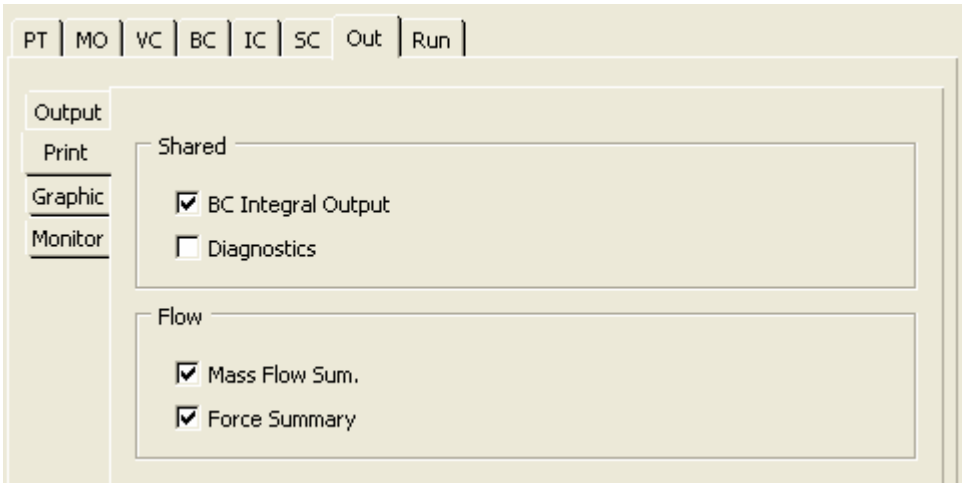

Figure 148. BC Integral Output Setup

The last step to take place in CFD-GUI is to launch the Optimization Manager under the Run tab:

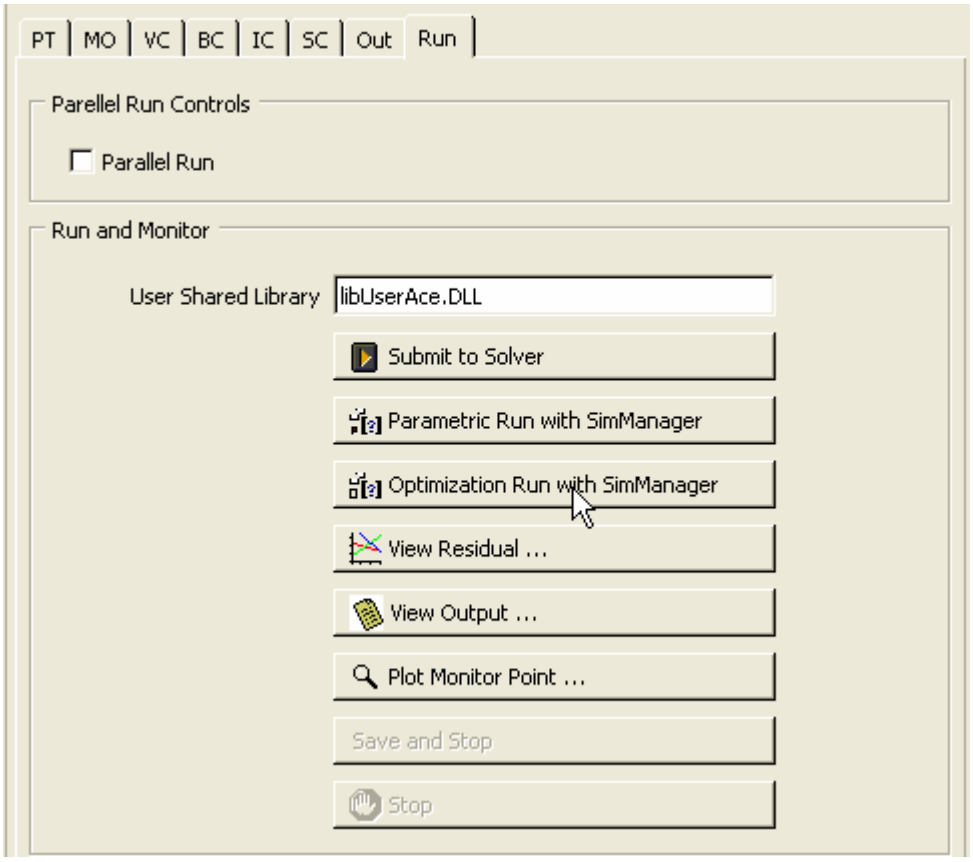

Figure 149. Launching Optimization Manager

f. Step Six– Setup the Optimization in the Optimization Manager

This portion of the setup is essentially an exercise in specifying the files created previously and setting constraints on the design variables. For this optimization, CFD-GEOM parameters were used with the Powell Multi Dimensional Optimizer. After checking the “Use CFD-GEOM parameters” check box the user must browse to find their variable geometry PYTHON file:

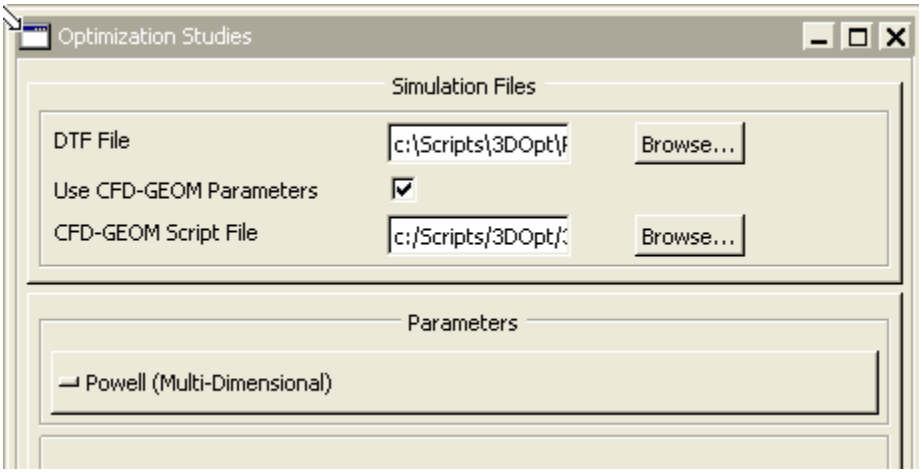

Figure 150. Specifying Script File and Optimizer

Just below this in the optimizer window are the parameters. In traditional optimization terminology these are considered the design variables. First the analyst must alter the BC button to reflect the fact that they are using GEOM parameters. After GEOM appears on the button, the analyst should specify the design variable and set the constraints on the design variables using the “Fill” button (see below):

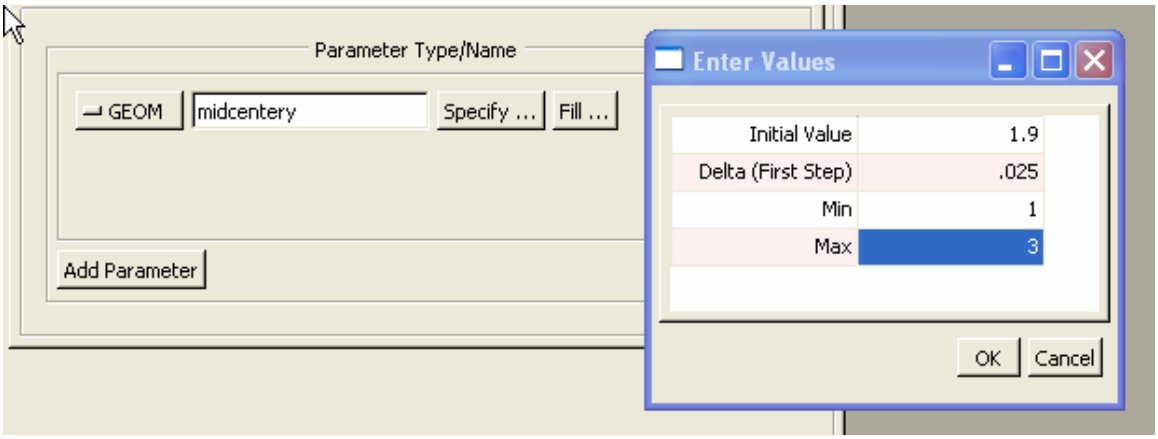

Figure 151. Setting Design Variables

The additional design variables should be added by pressing the add parameter button. It is important to note that in specifying initial values and step size that GEOM parameters are measured in meters unless a the user specifically changes the scale at the start of the entire process. The user should now hit the “next” button in order to access the last screen.

At this point the objective function will be specified for use by the optimizer. This is done by selecting “From User Script” as shown below:

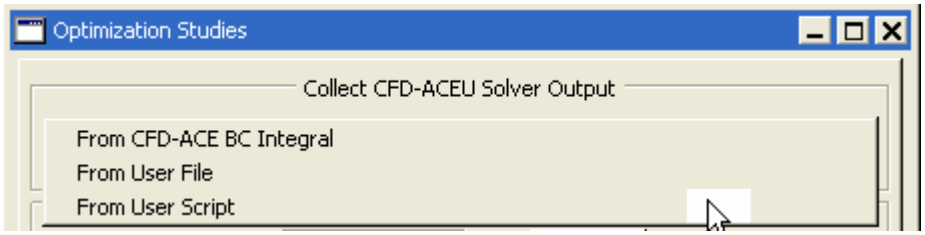

Figure 152. Setting Cost from User Script

The last selection to be made by the analyst is to browse for the PYTHON cost function file created in Step 4. For this thesis, a minimization was performed using the default optimizer settings. Clearly the analyst should also adjust these parameters at this point. Finally the analyst should simply select “Run All”. This will launch a graph that shows the cost function value versus iteration as well as a spreadsheet that does the same. See below:

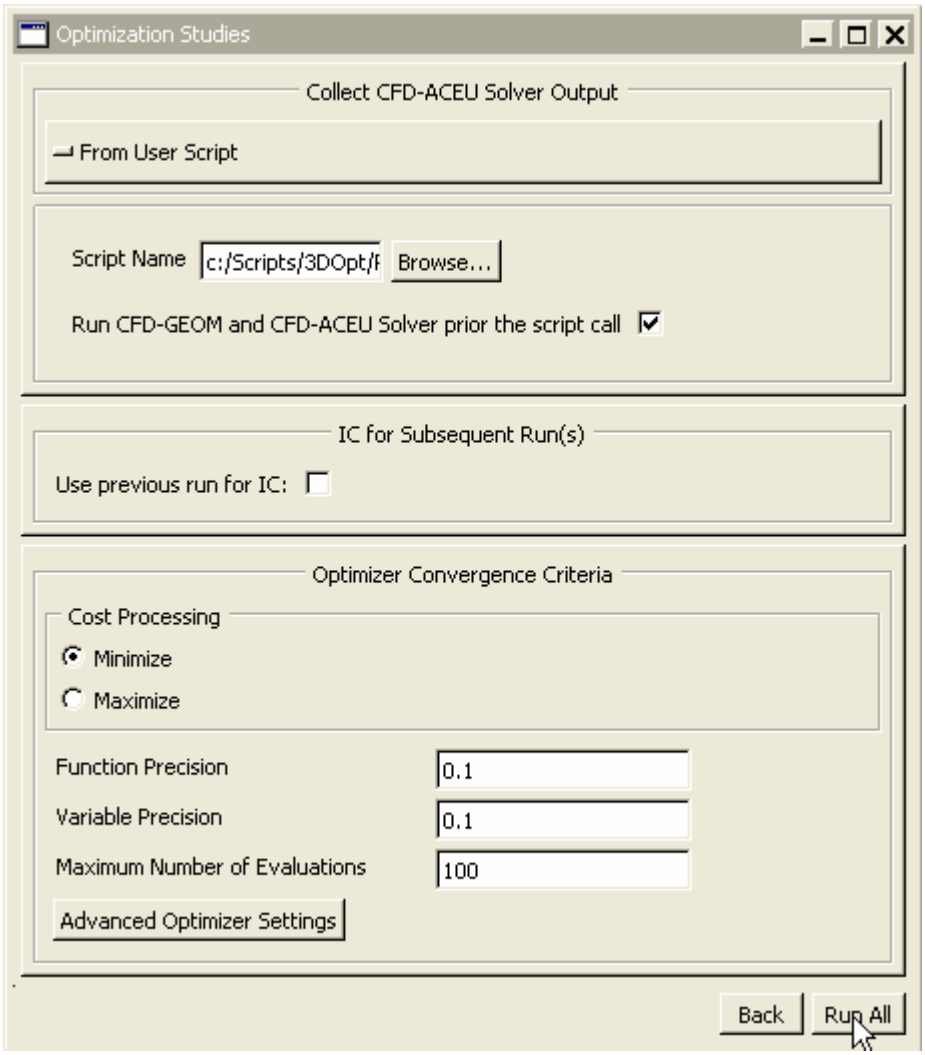

Figure 153. Launch Optimization

2. Two Dimensional Single Variable Optimization

A two dimensional single variable optimization was performed using the method detailed in this chapter. The only significant deviations are the optimization algorithm chosen and the number of design variables. In this case a single variable was used with the Brent optimization algorithm which is the equivalent of Powell's method in one dimension.

The Spline technique, discussed in chapter two, was used to develop the variable geometry. This essentially entailed placing a control point arbitrarily in the center of the canopy using CFD-GEOM with its height controlled by a design variable. Thus there was only a single variable to optimize. This simplified

approach was used to gain experience in optimization and control of the CFD optimizer. Placement of the design variable and its use is best understood by looking at the CFD-GEOM grid below:

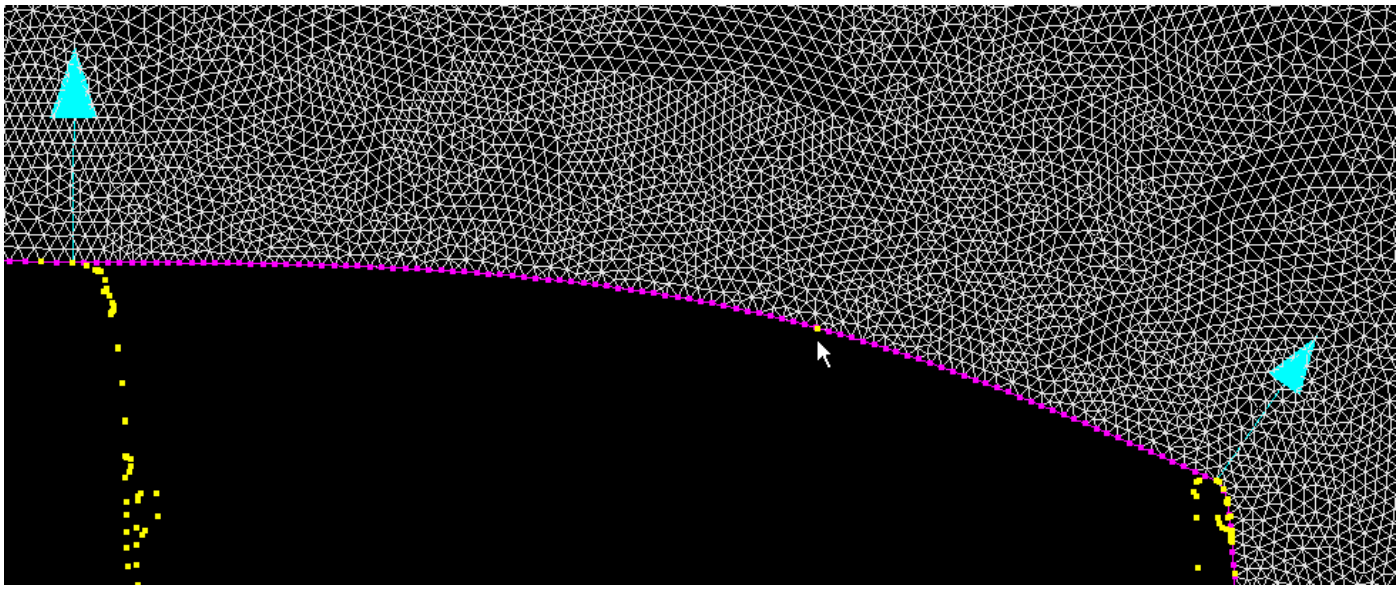

Figure 154. Unstructured Grid with Control Point Shown

The arrow is in the graphic above is pointing at the control point used for this optimization. The iterations performed by the optimizer in its search are shown below:

Iteration	Y (height)	Force n/unit depth
1	1.75	295.48
2	1.74	300.24
3	1.76	292.22
4	1.77	291.54
5	1.78	288.34
6	1.79	287.55
7	1.8	284.33
8	1.81	282.12
9	1.82	281.13
10	1.83	280.24
11	1.84	279.99
12	1.85	279.48
13	1.86	278.439
14	1.87	277.133
15	1.88	273.181
16	1.89	273.636
17	1.88397	273.847

Table 4. Two Dimensional Single Variable Optimization

Clearly the fifteenth iteration is a marked improvement over the initial design so the optimization was effective. Of course, an analyst with enough patience could have performed this optimization by simply creating each DTF file manually and continuously stepping the design variable until a local minimum was reached. This analysis was performed only to ensure that the software modules were functioning properly together and as a warm up for the multidimensional optimization. The optimizer truly adds value in multidimensional optimizations because it is able to select appropriate search directions without any input from the user. A more thorough analysis of the benefits of optimization will be provided in the next section as the three dimensional models are much more accurate at capturing the physics of the flow.

3. Three Dimensional Multi-Variable Optimization

The three dimensional optimization was performed using four design variables. The hybrid spline/domain element strategy was used. This strategy is explained in detail in chapter three. The optimization method selected was the Powell's multi variable method which is often called conjugate directions. Powell's method is considered to be one of the faster optimization algorithms, which was critical in this application. This is so important because each objective function call in three dimensions took over three hours. Four design variables were optimized so it also took a far greater number of function calls for the three dimensional optimization. Fortunately, Powell's method was able to find a local minimum in less than sixty objective function calls in all instances. Considering that there were nearly 60 objective function calls at three hours each the three dimensional optimizations took over a week each.

The variable geometry consisted of four control points. Two were located at the aft end of the truck and two were located midway up the canopy. As a result, three splines and variable length line were used to define three domain elements with all other domain sides remaining fixed. Examples of fixed sides, include the backside of the canopy, the bed rails and the tailgate opening. The

CFD-GEOM file is displayed graphically below (note: grids were turned off for clarity in visualization):

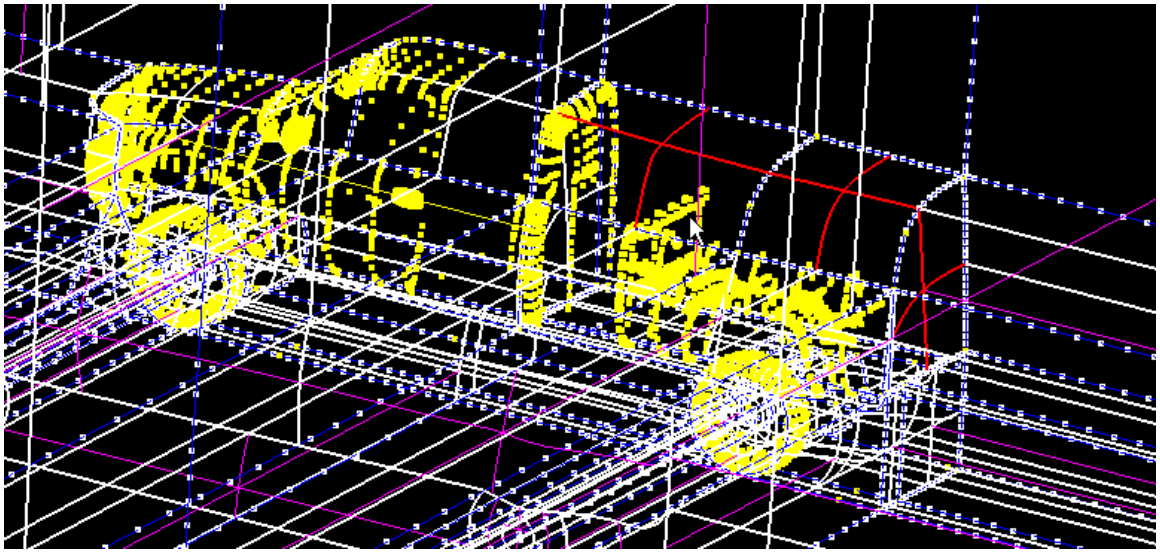

Figure 155. Three Dimensional Domain Elements

The three dimensional domain elements are the red faces (crosses) shown above. As mentioned previously, these domain elements geometrically morph as the design variables change with each optimization iteration. Another important construct behind this method are the splines. The splines are actually the building blocks for the curved domain elements. The three splines and the variable length line that were controlled by the design variables are highlighted in red below:

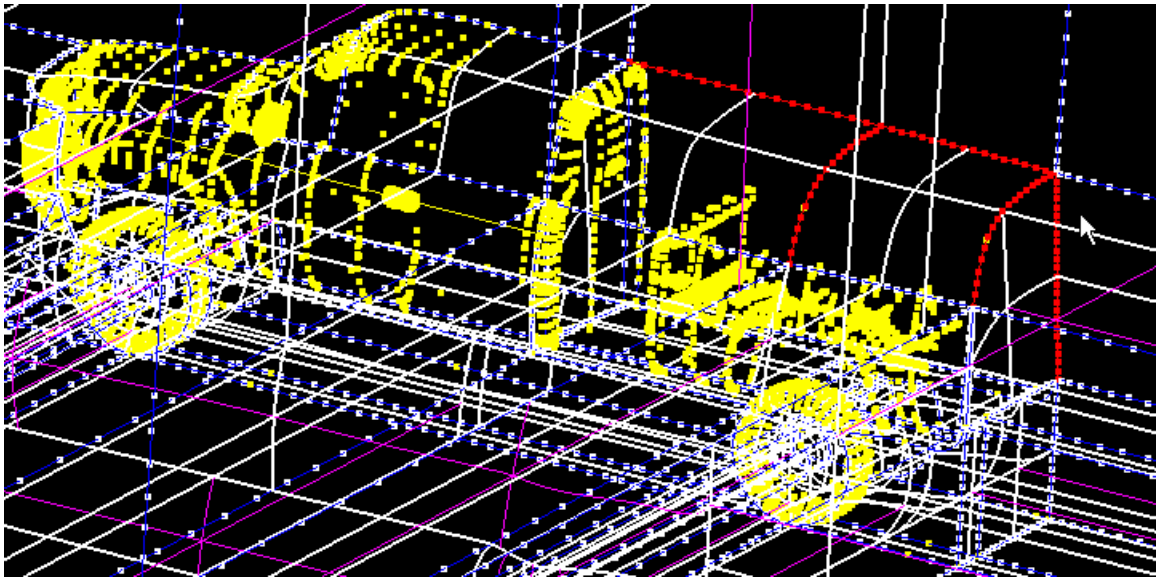

Figure 156. Control Splines and Variable Line

It should be obvious that much of the domain element remains fixed. The fixed sides are highlighted in red below:

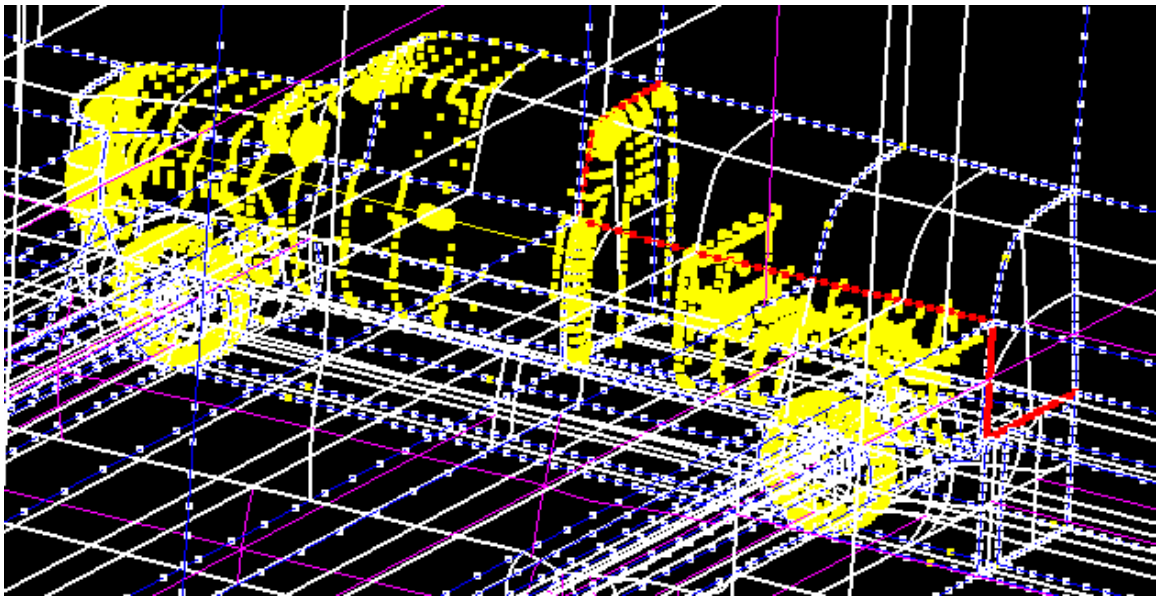

Figure 157. Fixed Sides of Domain Elements

In the case of this optimization it was fortunate that some of domain element sides were to remain fixed. Otherwise, the number of design variables necessary to create a flexible variable geometry would quickly skyrocket. As processing speed increases and becomes more readily available it is easy to see

that bodies can be developed with a completely flexible geometry that is only constrained by the volumes that must be enclosed.

The control points for the variable splines and line were varied only with respect to the Y direction. At first the author thought that the X and Z variables might also be helpful for maintaining flexibility in the variable geometry. A quick review of quadratic curves though brought the realization that a quadratic can be defined with just three variables. Adding more design variables to control the same point would be superfluous. The control points are shown on the next four figures in red with the pointer facing the subject point. They will be labeled individually so that the reader will be able to cross-reference them to the optimization data to follow:

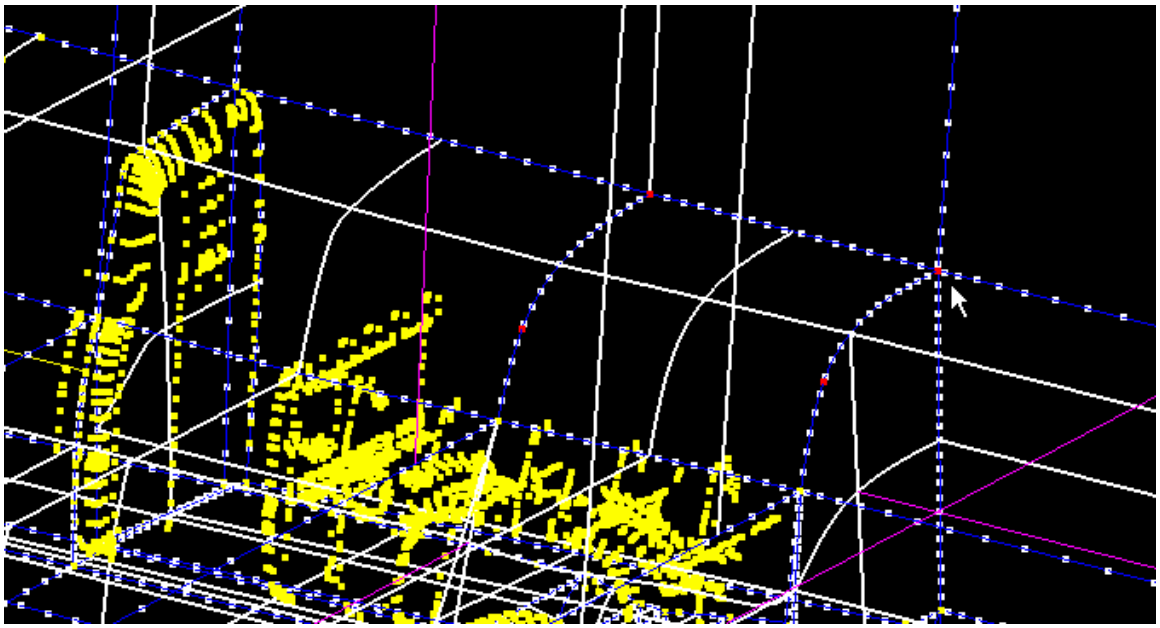

Figure 158. BackMove Control Point

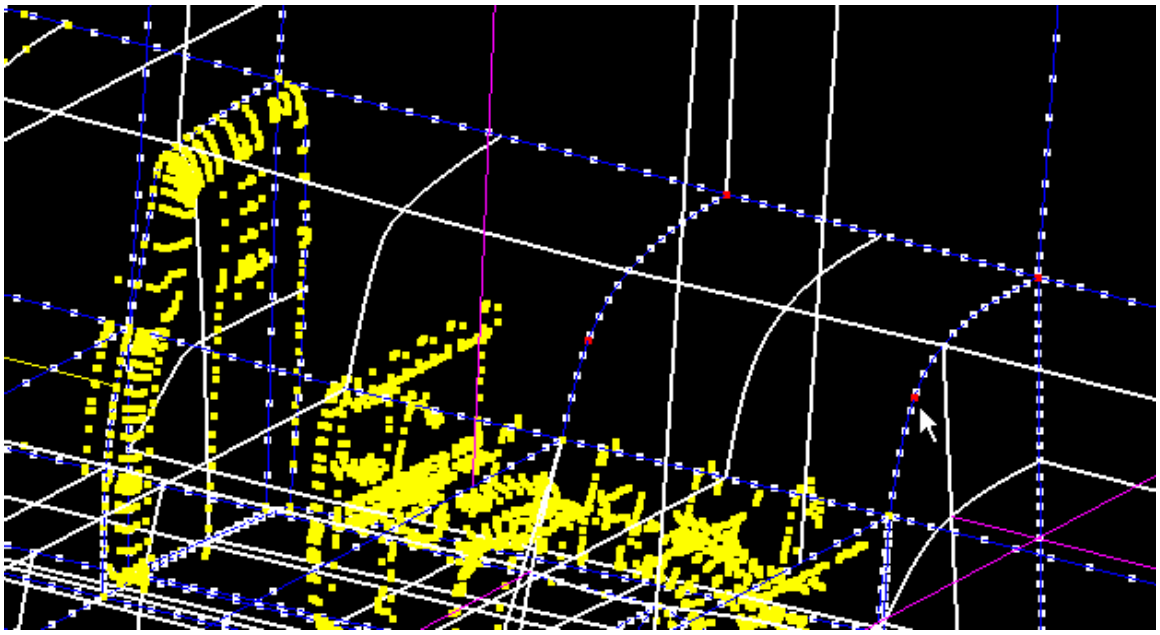

Figure 159. BackEdge Control Point

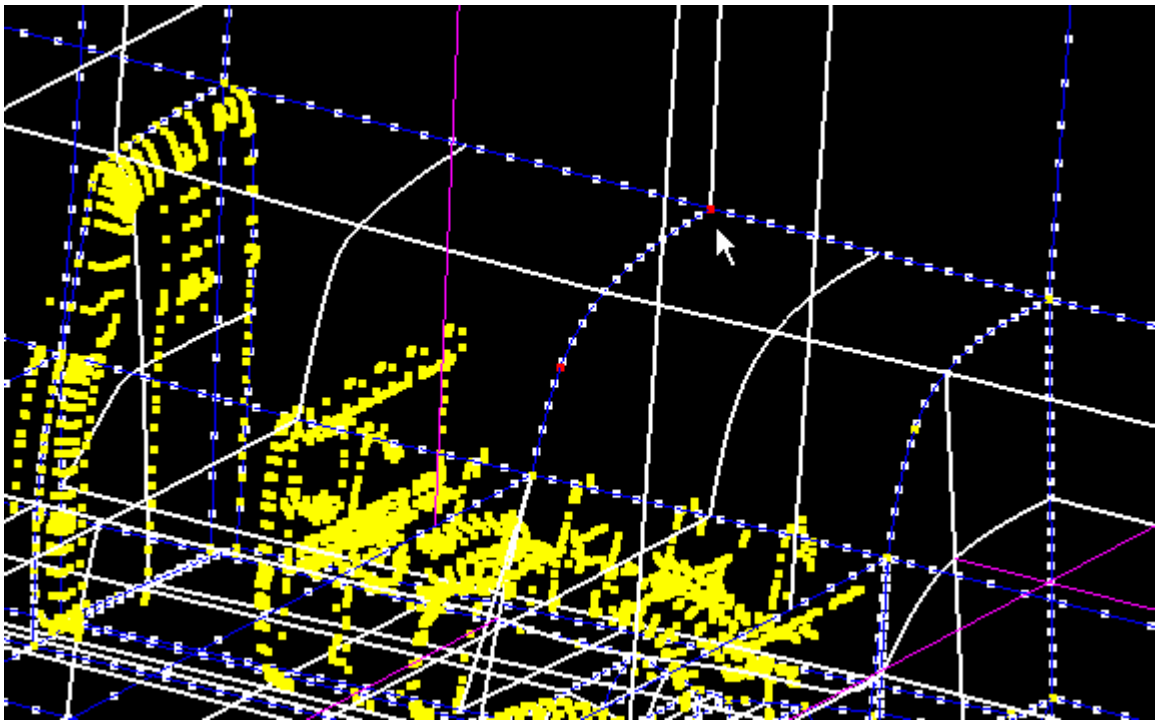

Figure 160. MidCenterY Control Point

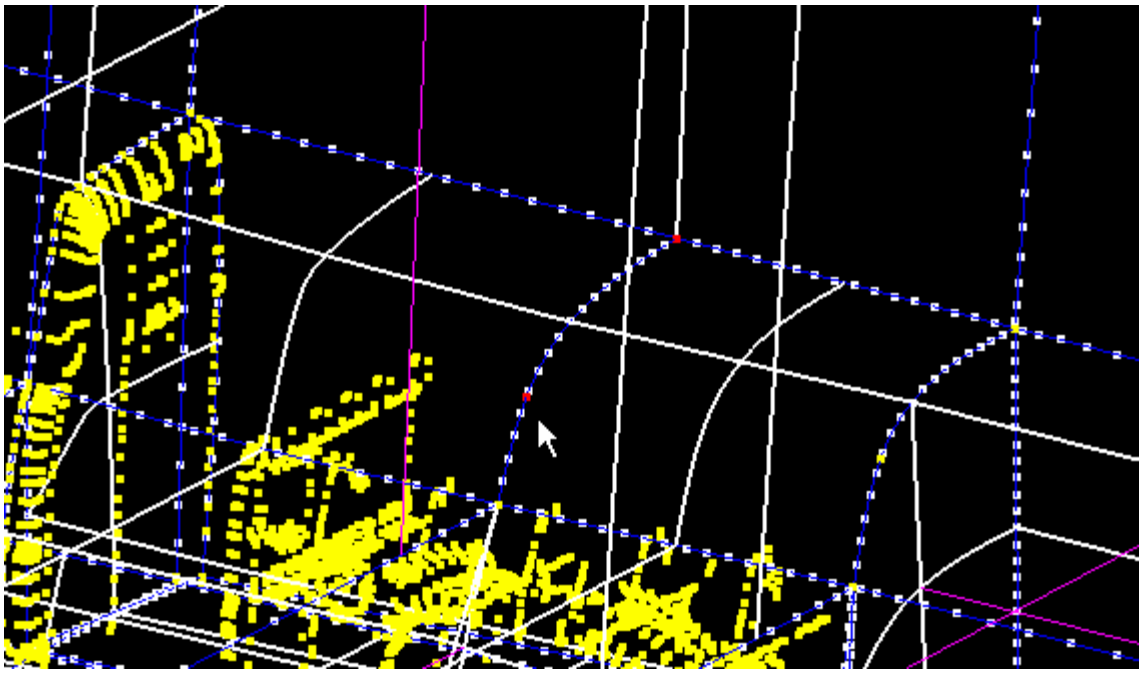

Figure 161. MidEdgeY Control Point

It was the intent of the author that the designs remain as flexible as possible while at the same time use a minimum number of control points. These four control points make virtually any quadratic shape possible for constructing the optimized canopy. The longitudinal spline is controlled by points MidCenterY and Backmove. The spline defining the tail domain is made up of the control points Backmove and BackEdge. Lastly, the spline crossing the canopy at the center is defined by the control points MidCenterY and MidEdgeY. As a result, the canopy height can taper along the length as well as the cross section. Many, if not all, aerodynamic shapes of interest should reside in this domain. Using this method, four optimization runs requiring one week of computation each were completed.

a. First Three Dimensional Optimization

The first three dimensional optimization run was started under the premise suggested by CFDRC [Ref. 11]. They suggest that the analyst start an optimization as near to the minimum as possible. This is a very logical argument considering the computational intensity of a three dimensional CFD optimization. The solved baseline configuration is shown below:

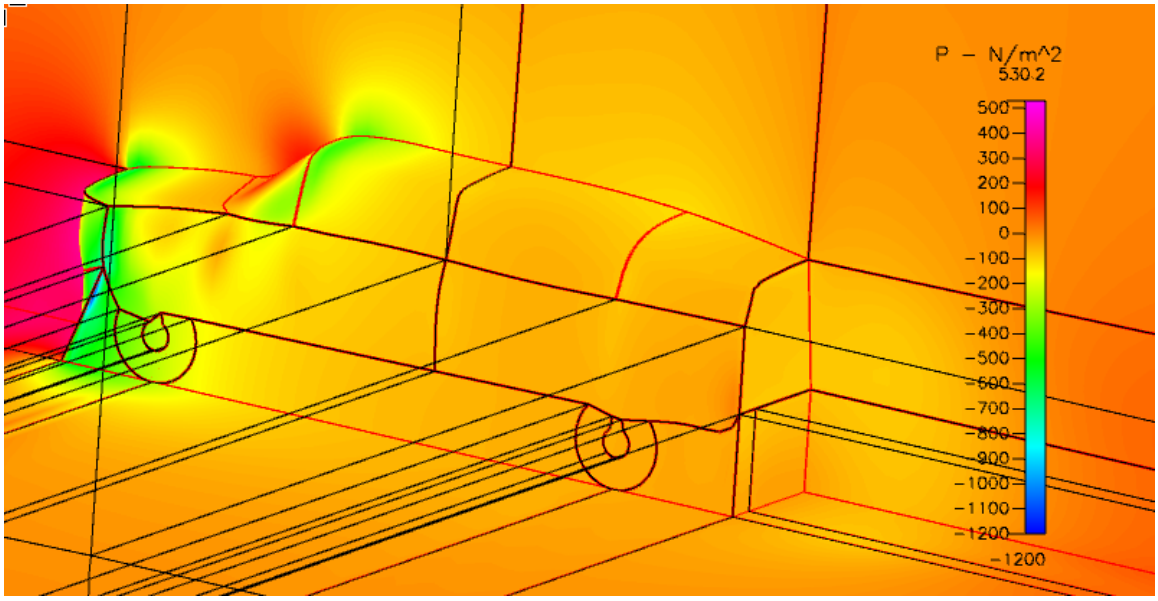

Figure 162. Optimization 1 – Iteration 1 Pressure Distribution

The curve along the centerline of the truck was designated based on the results of the two dimensional optimization. At this point, the two dimensional optimization was the "best guess" and it was the authors intent to find a local minimum as quickly as possible. Ironically, this optimization turned out to be the longest of the three in terms of iterations and computational time. It is important to note that it this model also employees the air dam, side skirts, and sealed rear area. This was done because it was found to be the best design in the previous section and the most likely to converge.

This first optimization required 57 iterations, the results of which are tabulated below:

Iteration	BackMove (m)	BackEdge (m)	MidEdgeY (m)	MidCenterY (m)	Drag (N)	% Change
1	1.770	1.682	1.700	1.925	427.793	32.90%
2	1.770	1.682	1.700	1.950	429.579	32.34%
3	1.770	1.682	1.700	1.975	431.555	31.74%
4	1.770	1.682	1.700	1.900	426.104	33.42%
5	1.770	1.682	1.700	1.875	424.742	33.85%
6	1.770	1.682	1.700	1.783	421.575	34.86%
7	1.770	1.682	1.700	1.783	421.575	34.86%
8	1.795	1.682	1.700	1.783	422.009	34.72%
9	1.820	1.682	1.700	1.783	422.815	34.46%
10	1.745	1.682	1.700	1.783	421.457	34.89%
11	1.720	1.682	1.700	1.783	421.786	34.79%

12	1.751	1.682	1.700	1.783	421.452	34.90%
13	1.751	1.682	1.700	1.783	421.452	34.90%
14	1.751	1.707	1.700	1.783	421.959	34.73%
15	1.751	1.732	1.700	1.783	422.541	34.55%
16	1.751	1.657	1.700	1.783	420.942	35.06%
17	1.751	1.632	1.700	1.783	420.597	35.17%
18	1.751	1.593	1.700	1.783	420.600	35.17%
19	1.751	1.593	1.700	1.783	420.600	35.17%
20	1.751	1.593	1.725	1.783	421.315	34.94%
21	1.751	1.593	1.750	1.783	422.194	34.66%
22	1.751	1.593	1.675	1.783	419.942	35.38%
23	1.751	1.593	1.650	1.783	419.483	35.53%
24	1.751	1.593	1.605	1.783	419.296	35.59%
25	1.751	1.593	1.605	1.783	419.296	35.59%
26	1.751	1.593	1.605	1.808	419.759	35.44%
27	1.751	1.593	1.605	1.833	420.441	35.22%
28	1.751	1.593	1.605	1.758	419.009	35.68%
29	1.751	1.593	1.605	1.733	419.055	35.67%
30	1.751	1.593	1.605	1.749	418.982	35.69%
31	1.751	1.593	1.605	1.749	418.982	35.69%
32	1.751	1.593	1.630	1.729	418.806	35.75%
33	1.751	1.593	1.655	1.709	418.859	35.73%
34	1.751	1.593	1.637	1.724	418.792	35.75%
35	1.751	1.593	1.637	1.724	418.792	35.75%
36	1.751	1.593	1.637	1.722	418.802	35.75%
37	1.751	1.593	1.637	1.720	418.816	35.74%
38	1.751	1.593	1.637	1.726	418.792	35.75%
39	1.751	1.593	1.637	1.728	418.795	35.75%
40	1.751	1.593	1.637	1.726	418.792	35.75%
41	1.751	1.593	1.637	1.726	418.792	35.75%
42	1.776	1.593	1.637	1.715	420.095	35.33%
43	1.801	1.593	1.637	1.705	422.309	34.62%
44	1.726	1.593	1.637	1.736	418.442	35.87%
45	1.701	1.593	1.637	1.746	418.554	35.83%
46	1.719	1.593	1.637	1.738	418.415	35.87%
47	1.719	1.593	1.637	1.738	418.415	35.87%
48	1.719	1.618	1.637	1.728	418.659	35.80%
49	1.719	1.643	1.637	1.717	419.185	35.63%
50	1.719	1.568	1.637	1.749	418.344	35.90%
51	1.719	1.543	1.637	1.760	418.434	35.87%
52	1.719	1.569	1.637	1.749	418.344	35.90%
53	1.719	1.569	1.637	1.749	418.344	35.90%
54	1.719	1.569	1.662	1.729	418.365	35.89%
55	1.719	1.569	1.687	1.709	418.518	35.84%
56	1.719	1.569	1.612	1.769	418.494	35.85%
57	1.719	1.569	1.646	1.741	418.337	35.90%

Table 5. Optimization 1 Data

(note: % comparison is based upon a stock truck with no aerodynamic enhancement)

The above data shows that a reasonable improvement can be obtained through a three dimensional CFD optimization. An improvement of 3% was observed when compared to the initial estimate of the optimal shape which was based upon the two dimensional optimum. All four design variables were tapered to some degree in order to reduce the drag. The bracketed minimum was captured at iteration 57 and its pressure distribution is shown below for comparison with the initial estimate:

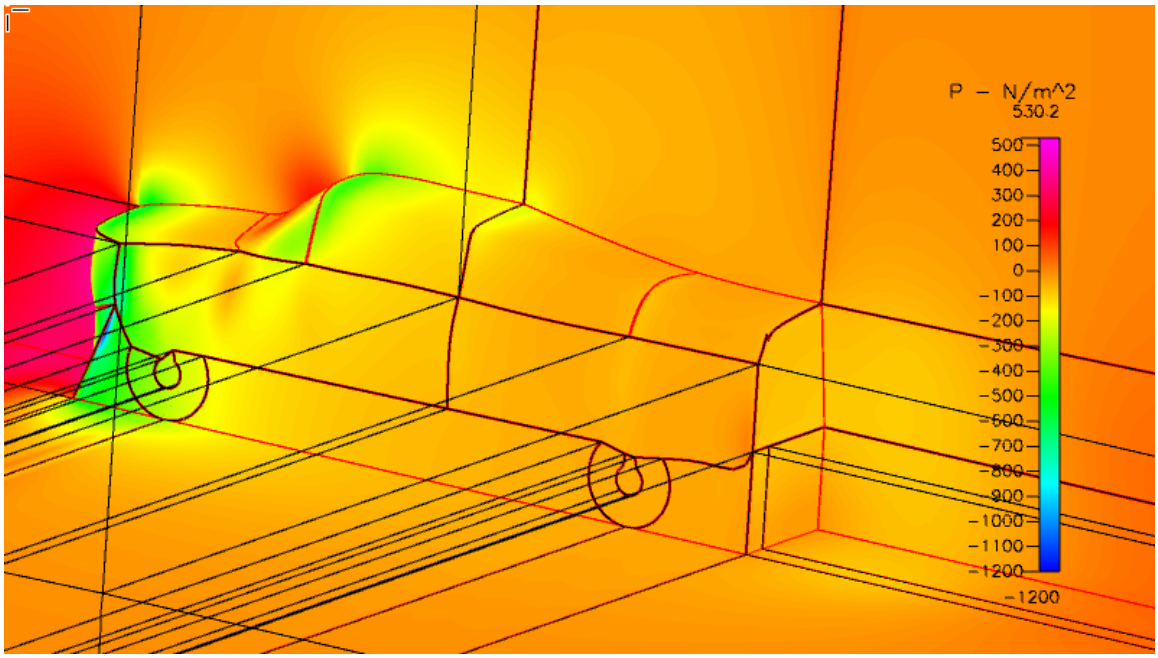

Figure 163. Optimization 1: Local Minimum Pressure Distribution

The most important difference is that the optimal geometry is actually concave (longitudinally) while the initial estimate was convex along all splines. This resulted in a smaller negative pressure at the trailing edge of the vehicle and therefore less drag. This result was not expected by the author and would probably never come about without the impartial optimization algorithm. The paradigm of the researcher may be one of the greatest hindrances to improvement and a powerful argument for using optimization.

Optimization without human influence clearly has its faults though. Iteration 53 is a repeat of iteration 52 as the design variables and cost function values are identical. Iteration 47 is a repeat of iteration 46 and the same goes for iterations: 35 and 34, 31 and 30, 24 and 25, and lastly 18 and 19. In total this

amounts to 18 hours of wasted computation. If there is a reason for calling the same design variables twice in a row, in any optimization algorithm, the author is unaware of it. This first optimization followed a very traditional path where the greatest gains are observed in the first several iterations and then flattens out as a local minimum is reached.

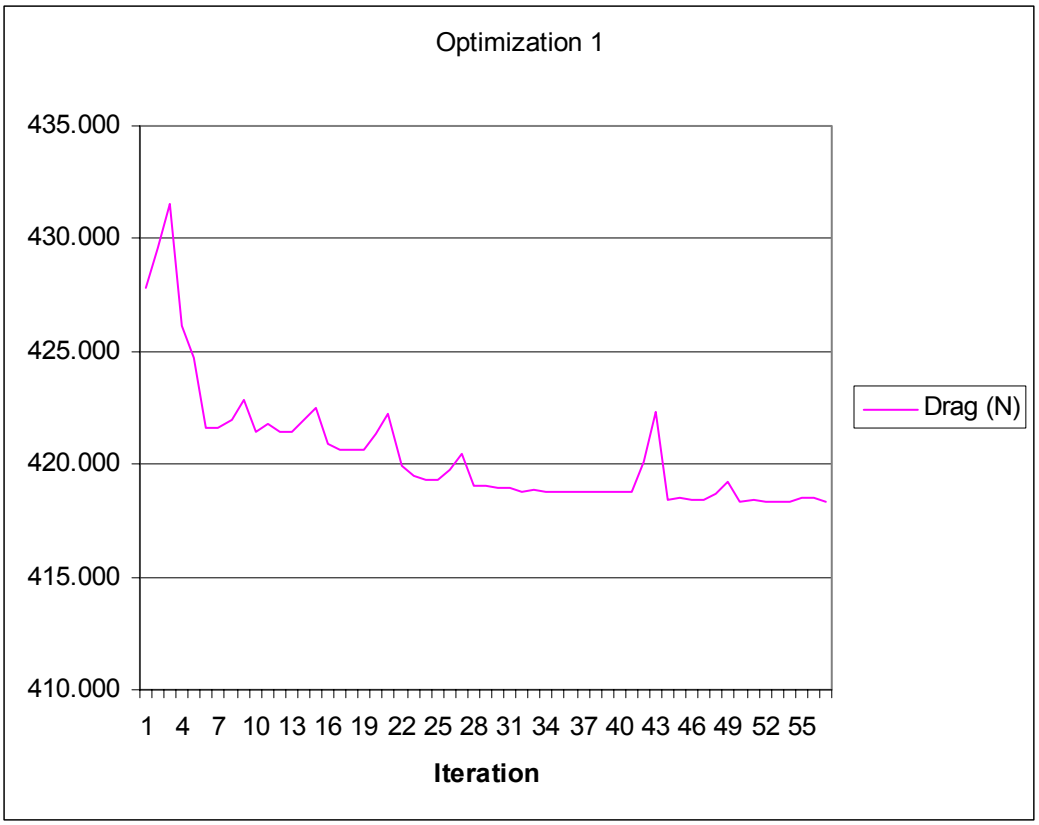

Figure 164. Optimization 1 Progression

b. Second Three Dimensional Optimization

The start point for this optimization was selected based upon the insight gained from the previous optimization. It was thought that a starting point with a much greater concavity would reduce its curvature to more closely mimic the local minimum observed in the first optimization. Again, intuition proved to be wrong. The start point for this optimization is displayed graphically below:

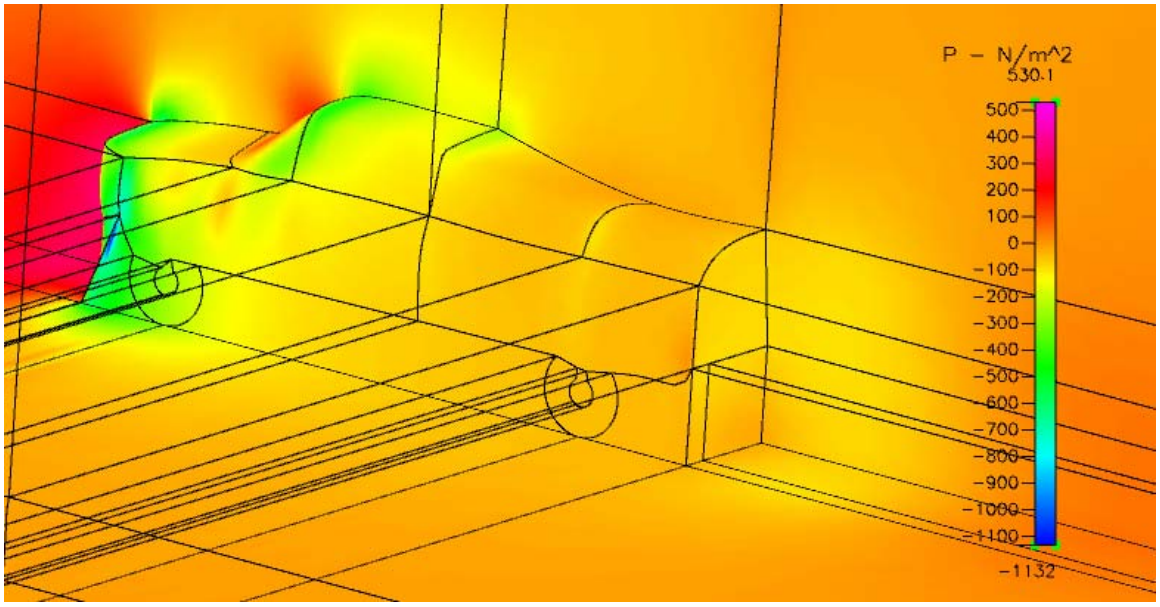

Figure 165. Optimization 2: Start Point

It was thought this solution would not change significantly as it was very near the end point of optimization 1 in the design space. It turned out that that the problem is not perfectly conditioned for optimization and that the first optimization had found a rather shallow local minimum at a 35.9% reduction in drag. The second optimization improved upon this figure considerable as it evolved to even greater concavity. The data representing the second optimization is shown in the table below:

Iteration	BackMove (m)	BackEdge (m)	MidEdgeY (m)	MidCenterY (m)	Drag (N)	% Change
1	1.55	1.479438	1.45	1.675	418.765	35.76%
2	1.55	1.479438	1.45	1.725	421.703	34.82%
3	1.55	1.479438	1.45	1.775	425.013	33.77%
4	1.55	1.479438	1.45	1.625	416.305	36.56%
5	1.55	1.479438	1.45	1.575	414.462	37.17%
6	1.55	1.479438	1.45	1.45064	412.441	37.84%
7	1.55	1.479438	1.45	1.45064	412.441	37.84%
8	1.6	1.479438	1.45	1.45064	414.271	37.23%
9	1.65	1.479438	1.45	1.45064	416.43	36.52%
10	1.5	1.479438	1.45	1.45064	411.729	38.08%
11	1.45	1.479438	1.45	1.45064	413.283	37.56%
12	1.50929	1.479438	1.45	1.45064	411.727	38.08%
13	1.50929	1.479438	1.45	1.45064	411.727	38.08%
14	1.50929	1.529438	1.45	1.45064	412.107	37.95%
15	1.50929	1.579438	1.45	1.45064	413.044	37.64%
16	1.50929	1.429438	1.45	1.45064	411.816	38.05%
17	1.50929	1.463895	1.45	1.45064	411.71	38.09%

18	1.50929	1.463895	1.45	1.45064	411.71	38.09%
19	1.50929	1.463895	1.5	1.45064	413.773	37.40%
20	1.50929	1.463895	1.55	1.45064	417.777	36.08%
21	1.50929	1.463895	1.4	1.45064	411.772	38.07%
22	1.50929	1.463895	1.42646	1.45064	411.447	38.18%
23	1.50929	1.463895	1.42646	1.45064	411.447	38.18%
24	1.50929	1.463895	1.42646	1.50064	411.755	38.07%
25	1.50929	1.463895	1.42646	1.55064	412.533	37.81%
26	1.50929	1.463895	1.42646	1.40064	411.927	38.01%
27	1.50929	1.463895	1.42646	1.45611	411.446	38.18%
28	1.50929	1.463895	1.42646	1.45611	411.446	38.18%
29	1.50929	1.463895	1.47646	1.40734	413.525	37.48%
30	1.50929	1.463895	1.52646	1.35858	420.106	35.33%
31	1.50929	1.463895	1.37646	1.50488	413.142	37.61%
32	1.50929	1.463895	1.42392	1.45859	411.464	38.17%
33	1.50929	1.463895	1.42392	1.45859	411.464	38.17%
34	1.50929	1.463895	1.42392	1.44952	411.458	38.17%
35	1.50929	1.463895	1.42392	1.44045	411.496	38.16%
36	1.50929	1.463895	1.42392	1.45279	411.455	38.17%
37	1.50929	1.463895	1.42392	1.45279	411.455	38.17%
38	1.55929	1.463895	1.42392	1.44932	413.435	37.51%
39	1.60929	1.463895	1.42392	1.44586	415.778	36.74%
40	1.45929	1.463895	1.42392	1.45625	411.068	38.30%
41	1.40929	1.463895	1.42392	1.45971	413.722	37.42%
42	1.47794	1.463895	1.42392	1.45496	410.946	38.34%
43	1.47794	1.463895	1.42392	1.45496	410.946	38.34%
44	1.47794	1.513895	1.42392	1.44971	411.288	38.23%
45	1.47794	1.563895	1.42392	1.44447	412.184	37.93%
46	1.47794	1.413895	1.42392	1.4602	411.075	38.30%
47	1.47794	1.452563	1.42392	1.45615	410.917	38.35%
48	1.47794	1.452563	1.42392	1.45615	410.917	38.35%
49	1.47794	1.452563	1.47392	1.40738	414.373	37.20%
50	1.47794	1.452563	1.52392	1.35861	422.588	34.53%
51	1.47794	1.452563	1.37392	1.50491	411.355	38.21%
52	1.47794	1.452563	1.40454	1.47504	410.689	38.43%

Table 6. Optimization 2 Data

The final shape is much more convex than was anticipated but improved the drag to the greatest extent of any optimization run. The graphical representation of the second local optimum is shown below:

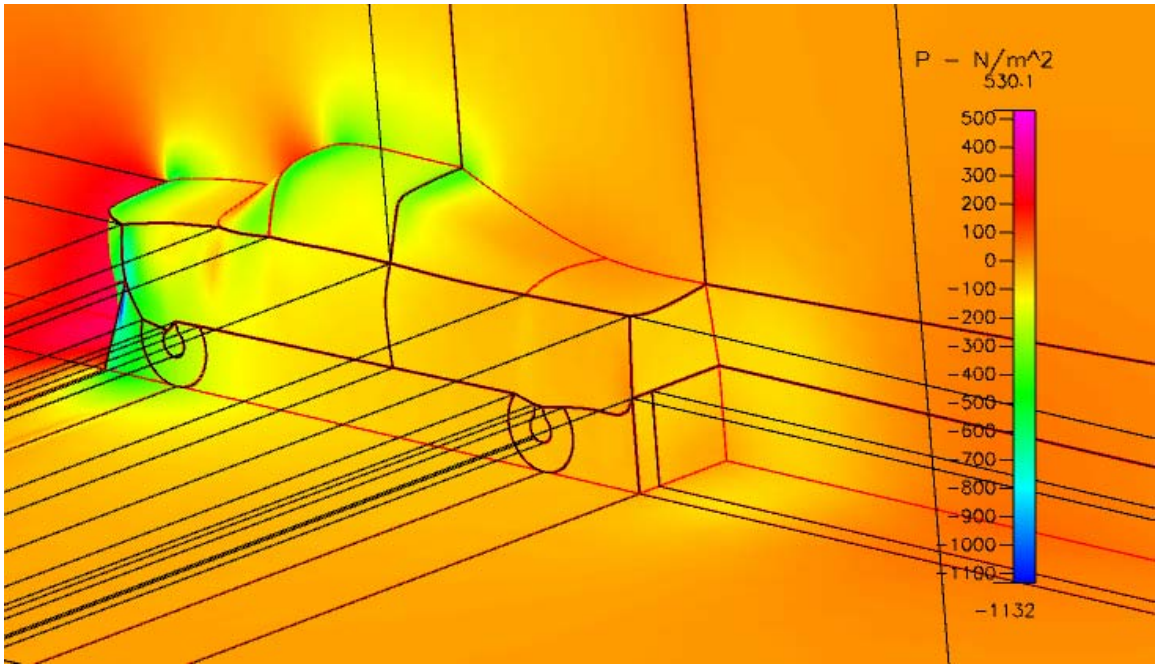

Figure 166. Optimization 2: Local Minimum

c. Third Three Dimensional Optimization

The third optimization was started to see whether a slightly higher concave canopy might be altered by the optimization algorithm and to see if it would converge to the same local minimum observed in optimization run 2. Yet again, the algorithm found its way to an entirely different local minimum that was larger in volume than the second optimization. The start point for this optimization is shown graphically below: (unfortunately this is only the grid due to corruption of the underlying dtf file)

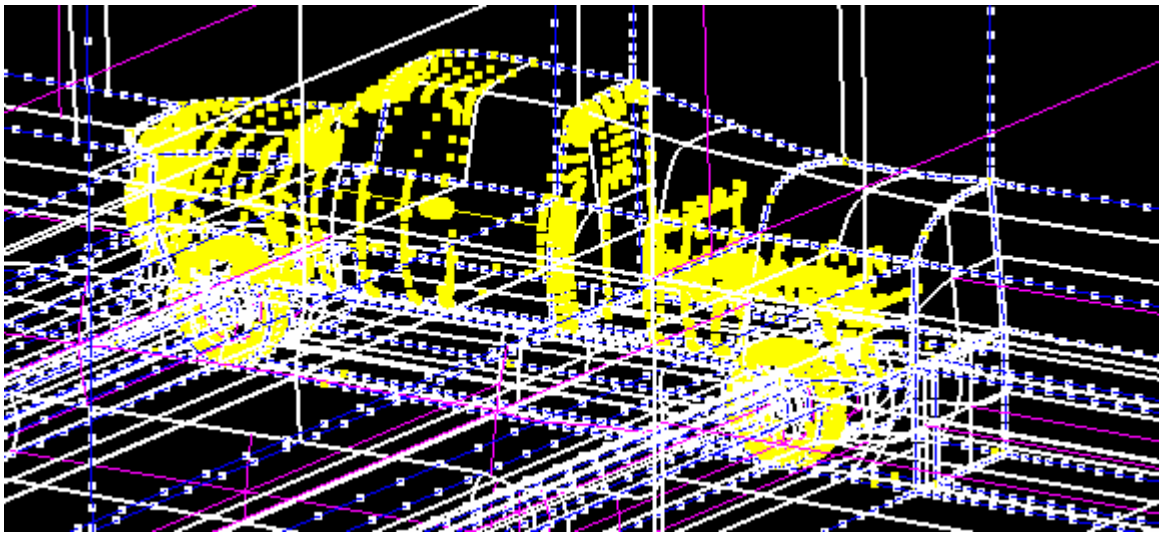

Figure 167. Optimization 3: Starting Point

Optimization 3 resulted in a slightly higher local minimum than optimization two. The results of this optimization are tabulated below:

Iteration	BackMove (m)	BackEdge (m)	MidEdgeY (m)	MidCenterY (m)	Drag (N)	% Change
1	1.646	1.728	1.570	1.691	416.889	36.37%
2	1.646	1.728	1.570	1.716	416.887	36.37%
3	1.646	1.728	1.570	1.741	417.253	36.25%
4	1.646	1.728	1.570	1.704	416.863	36.38%
5	1.596	1.728	1.570	1.704	416.457	36.51%
6	1.621	1.728	1.570	1.704	416.606	36.46%
7	1.646	1.728	1.570	1.704	416.863	36.38%
8	1.571	1.728	1.570	1.704	416.689	36.44%
9	1.599	1.728	1.570	1.704	416.465	36.51%
10	1.599	1.678	1.570	1.704	416.121	36.62%
11	1.599	1.703	1.570	1.704	416.224	36.59%
12	1.599	1.728	1.570	1.704	416.465	36.51%
13	1.599	1.653	1.570	1.704	416.134	36.62%
14	1.599	1.668	1.570	1.704	416.094	36.63%

15	1.599	1.668	1.520	1.704	415.405	36.86%
16	1.599	1.668	1.545	1.704	415.456	36.84%
17	1.599	1.668	1.570	1.704	416.094	36.63%
18	1.599	1.668	1.495	1.704	415.734	36.75%
19	1.599	1.668	1.529	1.704	415.371	36.87%
20	1.599	1.668	1.529	1.654	414.589	37.13%
21	1.599	1.668	1.529	1.679	414.838	37.05%
22	1.599	1.668	1.529	1.704	415.371	36.87%
23	1.599	1.668	1.529	1.629	414.620	37.12%
24	1.599	1.668	1.529	1.644	414.595	37.13%
25	1.599	1.668	1.479	1.681	415.519	36.82%
26	1.599	1.668	1.504	1.663	414.603	37.12%
27	1.599	1.668	1.529	1.644	414.595	37.13%
28	1.599	1.668	1.554	1.625	415.433	36.85%
29	1.599	1.668	1.517	1.653	414.412	37.19%
30	1.599	1.668	1.517	1.671	414.660	37.11%
31	1.599	1.668	1.517	1.662	414.510	37.15%
32	1.599	1.668	1.517	1.653	414.412	37.19%
33	1.599	1.668	1.517	1.644	414.351	37.21%
34	1.599	1.668	1.517	1.635	414.321	37.22%
35	1.599	1.668	1.517	1.631	414.319	37.22%
36	1.549	1.668	1.517	1.654	413.713	37.42%
37	1.574	1.668	1.517	1.643	413.956	37.34%
38	1.599	1.668	1.517	1.631	414.319	37.22%
39	1.524	1.668	1.517	1.666	413.708	37.42%
40	1.499	1.668	1.517	1.677	414.256	37.24%
41	1.536	1.668	1.517	1.660	413.660	37.44%
42	1.536	1.618	1.517	1.676	413.774	37.40%
43	1.536	1.643	1.517	1.668	413.612	37.45%
44	1.536	1.668	1.517	1.660	413.660	37.44%
45	1.536	1.650	1.517	1.666	413.604	37.46%
46	1.536	1.650	1.467	1.703	413.600	37.46%
47	1.536	1.650	1.492	1.685	413.099	37.62%
48	1.536	1.650	1.517	1.666	413.604	37.46%
49	1.536	1.650	1.492	1.685	413.100	37.62%

Table 7. Optimization 3 Data

The plot below shows that the optimization was a steady improvement from start to finish:

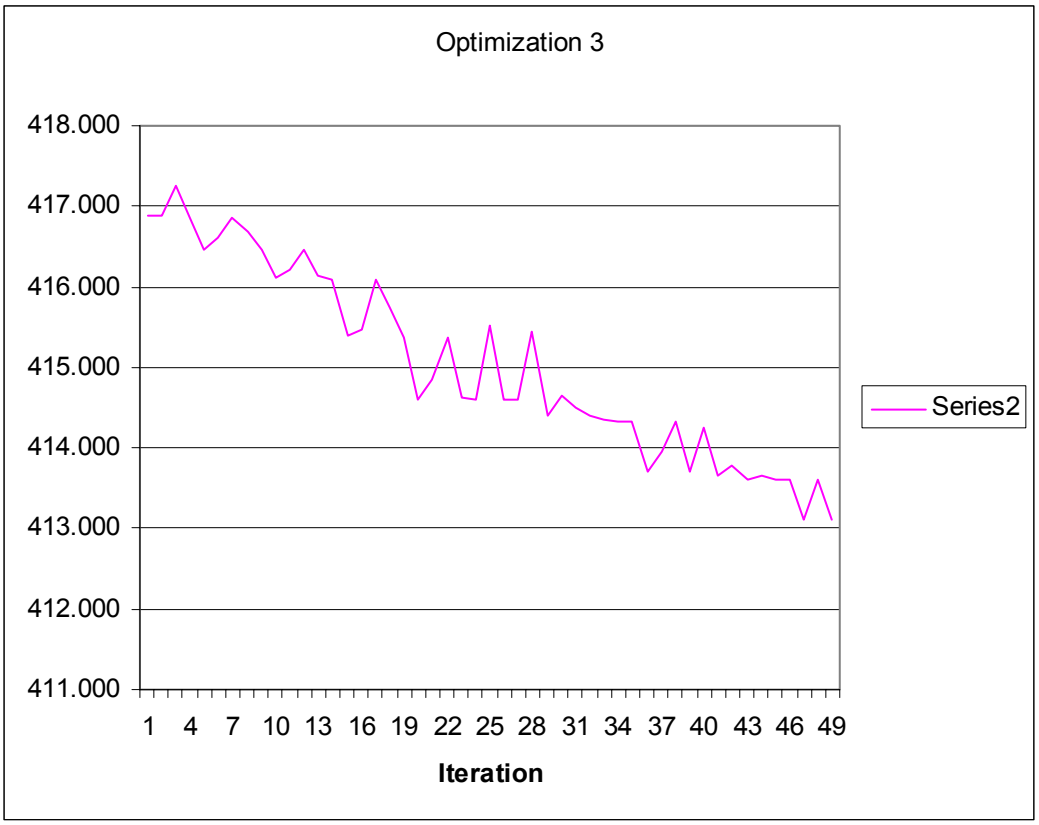

Figure 168. Optimization 3: Drag vs. Iteration

The final configuration of Optimization 3 was more similar to Optimization 1 but with performance similar to Optimization 2:

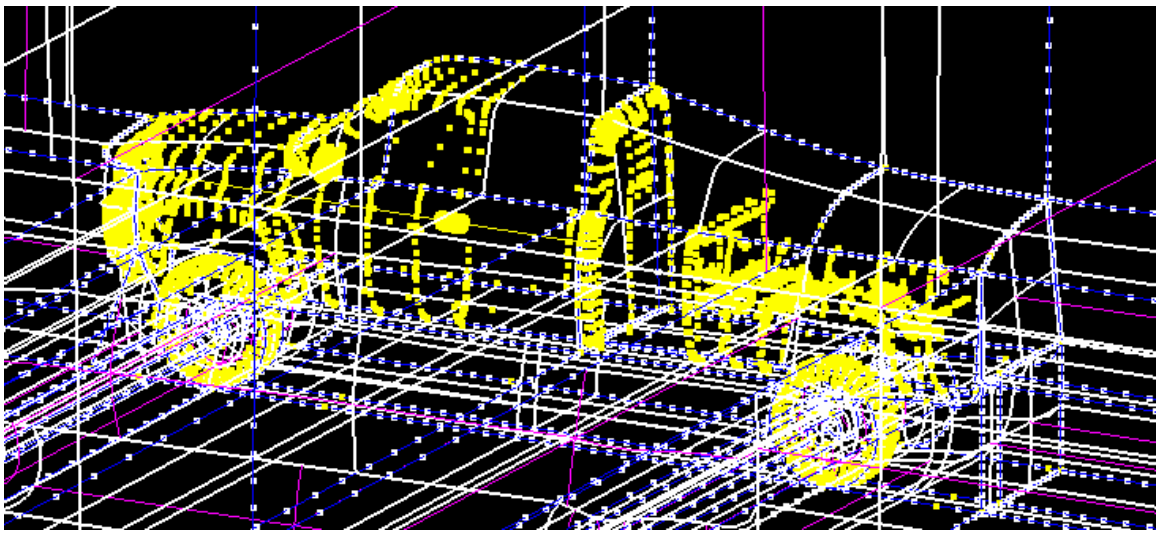

Figure 169. Optimization 3: Local Minimum

d. Fourth Three Dimensional Optimization

The last optimization began with the idea of improving the shape of a conventional canopy. The author wished to see if a traditional canopy optimization would result in a concave shape similar to the other optimizations or whether a local minimum that more nearly approximated the shape of a traditional canopy was present. To do this, the author started the optimization in the shape of a traditional canopy as depicted below:

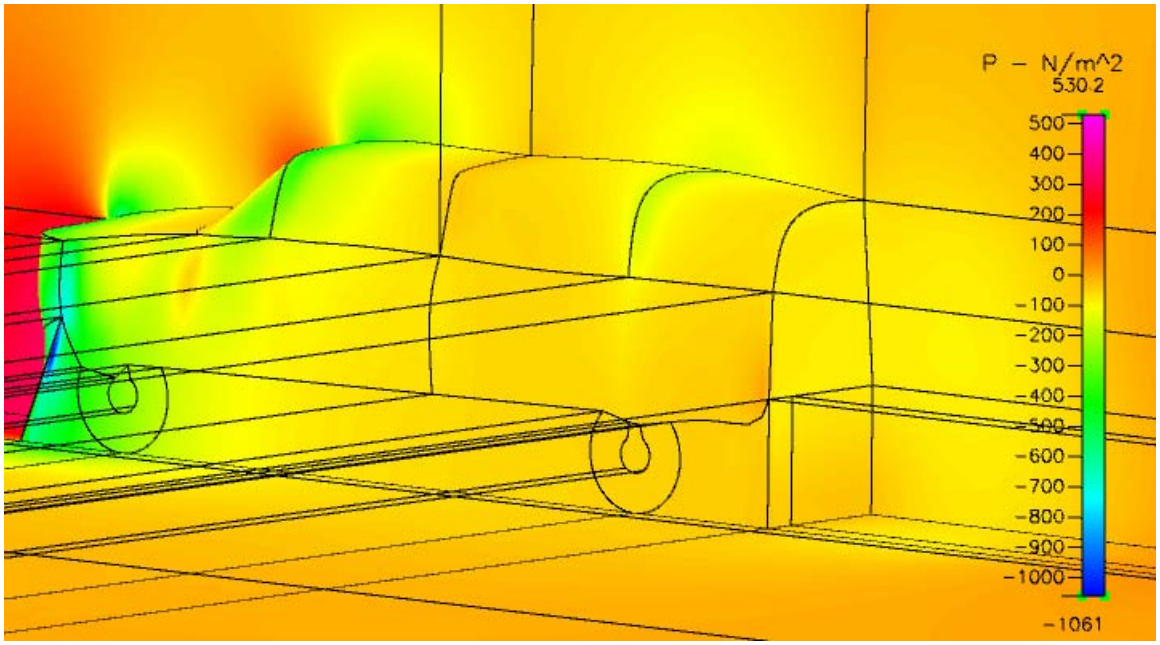

Figure 170. Optimization 4: Start Point

This optimization also converged to a unique local minimum that was significantly different than the first three optimizations. The data representing this process is tabulated below:

Iteration	BackMove (m)	BackEdge (m)	MidEdgeY (m)	MidCenterY (m)	Drag (N)	% Change
1	1.929438	1.8	1.9	2	433.66	31.10%
2	1.979438	1.85	1.9	2	434.633	30.80%
3	2.029438	1.9	1.9	2	439.704	29.30%
4	1.879438	1.75	1.9	2	435.301	30.60%
5	1.9358265	1.8063885	1.9	2	433.6	31.12%
6	1.9358265	1.8063885	1.8	1.9	432.317	31.51%
7	1.9358265	1.8063885	1.85	1.95	431.104	31.88%
8	1.9358265	1.8063885	1.9	2	433.6	31.12%
9	1.9358265	1.8063885	1.84135	1.94135	431.082	31.88%
10	1.9358265	1.7063885	1.84135	1.94135	428.252	32.75%

11	1.9358265	1.7563885	1.84135	1.94135	428.99	32.53%
12	1.9358265	1.8063885	1.84135	1.94135	431.082	31.88%
13	1.9358265	1.6563885	1.84135	1.94135	428.446	32.69%
14	1.9358265	1.6917885	1.84135	1.94135	428.141	32.79%
15	1.9358265	1.6917885	1.74135	1.94135	425.686	33.55%
16	1.9358265	1.6917885	1.79135	1.94135	426.628	33.26%
17	1.9358265	1.6917885	1.84135	1.94135	428.141	32.79%
18	1.9358265	1.6917885	1.69135	1.94135	425.565	33.59%
19	1.9358265	1.6917885	1.64135	1.94135	426.201	33.39%
20	1.9358265	1.6917885	1.708352	1.94135	425.551	33.60%
21	1.835827	1.591789	1.708352	1.94135	425.64	33.57%
22	1.8858265	1.6417885	1.708352	1.94135	424.204	34.02%
23	1.9358265	1.6917885	1.708352	1.94135	425.551	33.60%
24	1.8866263	1.6425883	1.708352	1.94135	424.195	34.02%
25	1.9477125	1.7036745	1.608352	1.94135	428.939	32.54%
26	1.9171694	1.6731314	1.658352	1.94135	424.51	33.92%
27	1.8866263	1.6425883	1.708352	1.94135	424.195	34.02%
28	1.856083	1.612045	1.758352	1.94135	426.869	33.18%
29	1.898685	1.654647	1.688611	1.94135	423.903	34.12%
30	1.923772	1.679734	1.688611	1.94135	424.83	33.82%
31	1.9112285	1.6671905	1.688611	1.94135	424.221	34.02%
32	1.898685	1.654647	1.688611	1.94135	423.903	34.12%
33	1.8861415	1.6421035	1.688611	1.94135	423.763	34.16%
34	1.873598	1.62956	1.688611	1.94135	423.775	34.16%
35	1.8808691	1.6368311	1.688611	1.94135	423.714	34.18%
36	1.9298874	1.6858494	1.588611	1.84135	432.045	31.59%
37	1.9053782	1.6613402	1.638611	1.89135	423.673	34.19%
38	1.8808691	1.6368311	1.688611	1.94135	423.714	34.18%
39	1.893243	1.649205	1.663371	1.91611	422.845	34.45%
40	1.9501317	1.6060937	1.663371	1.91611	422.112	34.68%
41	1.9216874	1.6276494	1.663371	1.91611	422.331	34.61%
42	1.893243	1.649205	1.663371	1.91611	422.845	34.45%
43	1.978576068	1.584538068	1.663371	1.91611	422.212	34.65%
44	1.9554229	1.6020849	1.663371	1.91611	422.101	34.69%
45	2.0165091	1.6631711	1.563371	1.91611	433.292	31.21%
46	1.98596604	1.63262804	1.613371	1.91611	424.858	33.81%
47	1.9554229	1.6020849	1.663371	1.91611	422.101	34.69%
48	1.9248798	1.5715418	1.713371	1.91611	423.29	34.31%
49	1.949355	1.596017	1.673304	1.91611	422.121	34.68%

Table 8. Optimization 4 Data

The fourth optimization produced similar gains from the initial design variable values as the other three though its local minimum was not as low. The optimization proceeded as shown in the figure below:

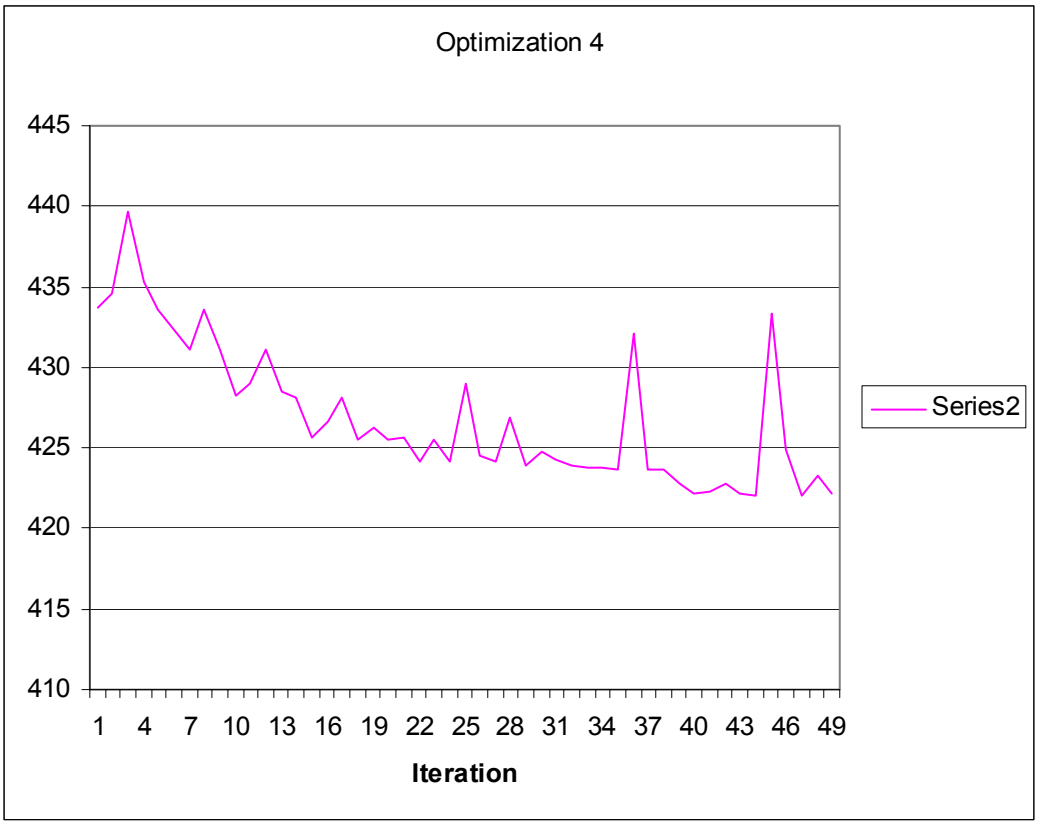

Figure 171. Optimization 4 - Drag vs. Iteration

The local minimum resulting from this optimization does not resemble any of the other three. Reasonably good gains in aerodynamic efficiency were seen in this optimization without forsaking as much cargo room. Unfortunately the author was unable to try more start points near the traditional canopy in order to find an even better compromise between efficiency and cargo room. The resulting local minimum from Optimization 4 is shown below:

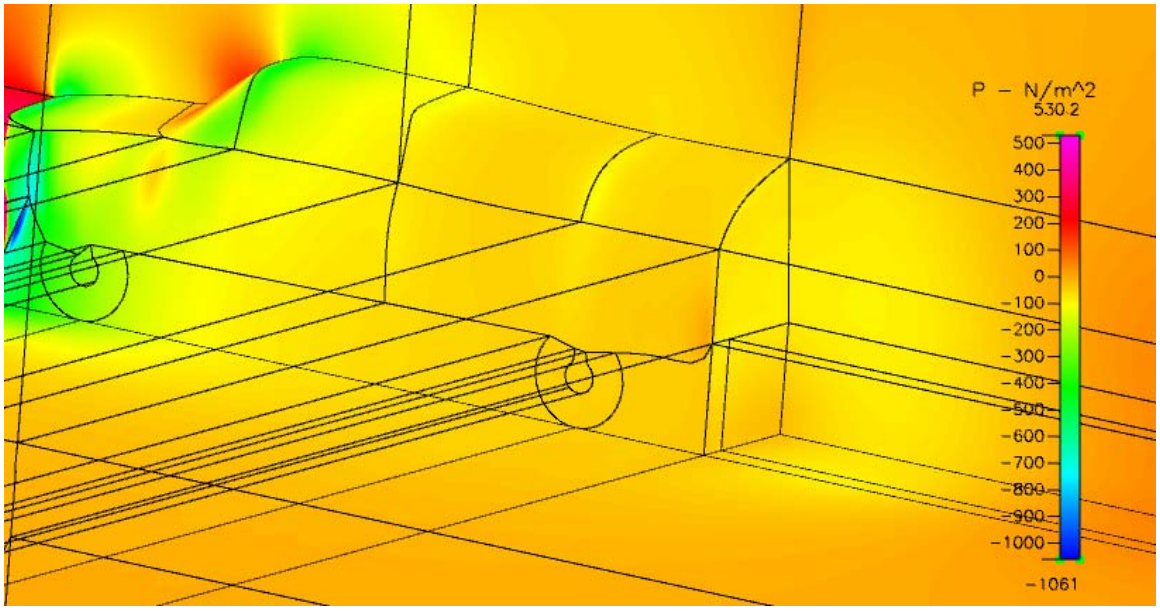

Figure 172. Optimization 4: Local Minimum

e. Selection of the Design

The preceding three optimizations produced local minimum that could be the optimal choice for an analyst depending upon their preference for cargo room. In this section, all optimization iterations will be compared for volume and a combination of design variables will be selected for the empirical study. To do this, the volume of the canopy was estimated using quadratic functions which are the basis of the 3 point spline used in the geometric definition of the canopy.

A simple method was employed in ranking the designs developed through the optimization iterations. Essentially, the author chose to give each design variable combination a score relative to the traditional canopy. First, this entailed dividing each design's volume by that of the traditional canopy. This became the volume score. Higher volumes are better and the largest volume is the traditional canopy.

The second score was based on drag, which was the primary focus of this research. To obtain these scores the author chose to divide the traditional canopy's drag by that of each of the designs. As a result, better designs (less drag) score higher which worked well with the higher scores generated by more voluminous designs. Since drag was the primary focus of this research the

author chose to weight the drag score double that of the volume score to make a combined score.

$$VolScore = \frac{Ivol}{Tvol}$$
$$Ivol = Iteration\ Volume$$
$$Tvol = Traditional\ Volume$$

$$DragScore = \frac{Tdrag}{Idrag}$$
$$Tdrag = Traditional\ Drag$$
$$Idrag = Iteration\ Drag$$

$$Overall = \frac{(2 * DragScore + VolScore)}{3}$$

Essentially the scores can be thought of as a percent difference from the baseline. The volume score is the percentage of volume out of the maximum possible. The drag score is the inverse of the percentage of the maximum represented in the field. Both scores are better as they get higher. The table below shows scores for all 4 optimization runs. The best score from optimization run 3 was selected as it was the best overall score although not the lowest drag. (Note: the 4th optimization is shown first because the first iteration is the baseline for comparison.)

4th Optimization Results				
Iteration	tot Vol	Vol Score	DragScore	Combined
1	2.6788	1.0000	1.0000	1.0000
2	2.7114	1.0122	0.9906	0.9978
3	2.7440	1.0243	0.9421	0.9695
4	2.6461	0.9878	0.9841	0.9853
5	2.6829	1.0016	1.0006	1.0009
6	2.5387	0.9477	1.0131	0.9913
7	2.6108	0.9746	1.0250	1.0082
8	2.6829	1.0016	1.0006	1.0009
9	2.5983	0.9700	1.0252	1.0068
10	2.5474	0.9510	1.0532	1.0191
11	2.5729	0.9605	1.0459	1.0174

12	2.5983	0.9700	1.0252	1.0068
13	2.5219	0.9415	1.0513	1.0147
14	2.5400	0.9482	1.0543	1.0190
15	2.4296	0.9070	1.0790	1.0216
16	2.4848	0.9276	1.0695	1.0222
17	2.5400	0.9482	1.0543	1.0190
18	2.3744	0.8864	1.0802	1.0156
19	2.3192	0.8658	1.0738	1.0044
20	2.3931	0.8934	1.0803	1.0180
21	2.3279	0.8690	1.0794	1.0093
22	2.3605	0.8812	1.0940	1.0230
23	2.3931	0.8934	1.0803	1.0180
24	2.3610	0.8814	1.0941	1.0232
25	2.2905	0.8551	1.0464	0.9826
26	2.3258	0.8682	1.0909	1.0167
27	2.3610	0.8814	1.0941	1.0232
28	2.3963	0.8946	1.0671	1.0096
29	2.3471	0.8762	1.0970	1.0234
30	2.3635	0.8823	1.0876	1.0192
31	2.3553	0.8792	1.0938	1.0223
32	2.3471	0.8762	1.0970	1.0234
33	2.3389	0.8731	1.0985	1.0234
34	2.3307	0.8701	1.0983	1.0223
35	2.3355	0.8719	1.0990	1.0233
36	2.2232	0.8299	1.0158	0.9538
37	2.2793	0.8509	1.0994	1.0165
38	2.3355	0.8719	1.0990	1.0233
39	2.3071	0.8613	1.1078	1.0256
40	2.2933	0.8561	1.1153	1.0289
41	2.3002	0.8587	1.1131	1.0283
42	2.3071	0.8613	1.1078	1.0256
43	2.2864	0.8535	1.1143	1.0274
44	2.2920	0.8556	1.1154	1.0288
45	2.2215	0.8293	1.0036	0.9455
46	2.2568	0.8425	1.0873	1.0057
47	2.2920	0.8556	1.1154	1.0288
48	2.3273	0.8688	1.1033	1.0251
49	2.2990	0.8583	1.1152	1.0296
1st Optimization Results				
Iteration	tot Vol	Vol Score	DragScore	Combined
1	2.3498	0.8772	1.0578	0.9976
2	2.3583	0.8804	1.0400	0.9868
3	2.3667	0.8835	1.0206	0.9749
4	2.3414	0.8740	1.0748	1.0079
5	2.3329	0.8709	1.0885	1.0160
6	2.3019	0.8593	1.1208	1.0337

7	2.3019	0.8593	1.1208	1.0337
8	2.3055	0.8606	1.1164	1.0311
9	2.3090	0.8620	1.1081	1.0261
10	2.2983	0.8580	1.1221	1.0340
11	2.2947	0.8566	1.1187	1.0313
12	2.2992	0.8583	1.1221	1.0342
13	2.2992	0.8583	1.1221	1.0342
14	2.3119	0.8630	1.1169	1.0323
15	2.3246	0.8678	1.1109	1.0299
16	2.2864	0.8535	1.1274	1.0361
17	2.2737	0.8488	1.1309	1.0369
18	2.2537	0.8413	1.1309	1.0344
19	2.2537	0.8413	1.1309	1.0344
20	2.2813	0.8516	1.1235	1.0329
21	2.3089	0.8619	1.1145	1.0303
22	2.2261	0.8310	1.1377	1.0355
23	2.1985	0.8207	1.1425	1.0352
24	2.1487	0.8021	1.1444	1.0303
25	2.1487	0.8021	1.1444	1.0303
26	2.1572	0.8053	1.1396	1.0282
27	2.1656	0.8084	1.1325	1.0245
28	2.1402	0.7990	1.1474	1.0313
29	2.1318	0.7958	1.1469	1.0299
30	2.1372	0.7978	1.1477	1.0311
31	2.1372	0.7978	1.1477	1.0311
32	2.1580	0.8056	1.1495	1.0349
33	2.1788	0.8134	1.1490	1.0371
34	2.1636	0.8077	1.1497	1.0357
35	2.1636	0.8077	1.1497	1.0357
36	2.1628	0.8074	1.1496	1.0355
37	2.1621	0.8071	1.1494	1.0353
38	2.1643	0.8079	1.1497	1.0358
39	2.1650	0.8082	1.1496	1.0358
40	2.1641	0.8079	1.1497	1.0357
41	2.1641	0.8079	1.1497	1.0357
42	2.1642	0.8079	1.1361	1.0267
43	2.1643	0.8080	1.1133	1.0115
44	2.1640	0.8078	1.1533	1.0382
45	2.1638	0.8078	1.1521	1.0374
46	2.1639	0.8078	1.1536	1.0383
47	2.1639	0.8078	1.1536	1.0383
48	2.1730	0.8112	1.1511	1.0378
49	2.1821	0.8146	1.1456	1.0352
50	2.1549	0.8044	1.1543	1.0377
51	2.1458	0.8010	1.1534	1.0359
52	2.1554	0.8046	1.1543	1.0378
53	2.1554	0.8046	1.1543	1.0378
54	2.1762	0.8124	1.1541	1.0402

55	2.1970	0.8202	1.1525	1.0417
56	2.1345	0.7968	1.1528	1.0341
57	2.1633	0.8076	1.1544	1.0388
2nd Optimization Results				
Iteration	tot Vol	Vol Score	DragScore	Combined
1	1.8544	0.6923	1.1499	0.9974
2	1.8714	0.6986	1.1195	0.9792
3	1.8883	0.7049	1.0858	0.9588
4	1.8375	0.6859	1.1757	1.0125
5	1.8206	0.6796	1.1953	1.0234
6	1.7785	0.6639	1.2169	1.0326
7	1.7785	0.6639	1.2169	1.0326
8	1.7856	0.6666	1.1973	1.0204
9	1.7927	0.6692	1.1744	1.0060
10	1.7713	0.6612	1.2245	1.0368
11	1.7642	0.6586	1.2079	1.0248
12	1.7726	0.6617	1.2246	1.0370
13	1.7726	0.6617	1.2246	1.0370
14	1.7981	0.6712	1.2205	1.0374
15	1.8236	0.6807	1.2104	1.0339
16	1.7472	0.6522	1.2236	1.0332
17	1.7647	0.6588	1.2248	1.0361
18	1.7647	0.6588	1.2248	1.0361
19	1.8199	0.6794	1.2026	1.0282
20	1.8751	0.7000	1.1603	1.0068
21	1.7095	0.6382	1.2241	1.0288
22	1.7387	0.6491	1.2276	1.0348
23	1.7387	0.6491	1.2276	1.0348
24	1.7557	0.6554	1.2243	1.0346
25	1.7726	0.6617	1.2159	1.0312
26	1.7218	0.6428	1.2224	1.0292
27	1.7406	0.6498	1.2276	1.0350
28	1.7406	0.6498	1.2276	1.0350
29	1.7793	0.6642	1.2053	1.0249
30	1.8180	0.6787	1.1360	0.9836
31	1.7019	0.6353	1.2094	1.0180
32	1.7386	0.6490	1.2274	1.0346
33	1.7386	0.6490	1.2274	1.0346
34	1.7355	0.6479	1.2275	1.0343
35	1.7325	0.6467	1.2271	1.0336
36	1.7367	0.6483	1.2275	1.0344
37	1.7367	0.6483	1.2275	1.0344
38	1.7426	0.6505	1.2062	1.0210
39	1.7486	0.6528	1.1813	1.0051
40	1.7307	0.6461	1.2317	1.0365
41	1.7247	0.6438	1.2032	1.0167

42	1.7329	0.6469	1.2330	1.0376
43	1.7329	0.6469	1.2330	1.0376
44	1.7566	0.6557	1.2293	1.0381
45	1.7803	0.6646	1.2196	1.0346
46	1.7092	0.6381	1.2316	1.0338
47	1.7275	0.6449	1.2333	1.0372
48	1.7275	0.6449	1.2333	1.0372
49	1.7662	0.6593	1.1962	1.0173
50	1.8049	0.6738	1.1105	0.9649
51	1.6888	0.6305	1.2286	1.0292
52	1.7125	0.6393	1.2358	1.0370
3rd Optimization Results				
Iteration	tot Vol	Vol Score	DragScore	Combined
1	2.1326	0.7961	1.1696	1.0451
2	2.1411	0.7993	1.1696	1.0462
3	2.1496	0.8025	1.1658	1.0447
4	2.1369	0.7977	1.1699	1.0458
5	2.1298	0.7951	1.1741	1.0478
6	2.1333	0.7964	1.1726	1.0472
7	2.1369	0.7977	1.1699	1.0458
8	2.1262	0.7937	1.1717	1.0457
9	2.1302	0.7952	1.1741	1.0478
10	2.1047	0.7857	1.1777	1.0470
11	2.1174	0.7905	1.1766	1.0479
12	2.1302	0.7952	1.1741	1.0478
13	2.0920	0.7809	1.1775	1.0453
14	2.0997	0.7838	1.1780	1.0466
15	2.0445	0.7632	1.1853	1.0446
16	2.0721	0.7735	1.1847	1.0477
17	2.0997	0.7838	1.1780	1.0466
18	2.0169	0.7529	1.1818	1.0388
19	2.0546	0.7670	1.1856	1.0461
20	2.0377	0.7607	1.1939	1.0495
21	2.0461	0.7638	1.1913	1.0488
22	2.0546	0.7670	1.1856	1.0461
23	2.0292	0.7575	1.1936	1.0482
24	2.0344	0.7595	1.1939	1.0491
25	1.9918	0.7436	1.1840	1.0372
26	2.0131	0.7515	1.1938	1.0463
27	2.0344	0.7595	1.1939	1.0491
28	2.0557	0.7674	1.1850	1.0458
29	2.0239	0.7556	1.1958	1.0491
30	2.0301	0.7578	1.1932	1.0481
31	2.0270	0.7567	1.1948	1.0487
32	2.0239	0.7556	1.1958	1.0491
33	2.0209	0.7544	1.1965	1.0491

34	2.0178	0.7533	1.1968	1.0489
35	2.0165	0.7528	1.1968	1.0488
36	2.0172	0.7530	1.2033	1.0532
37	2.0169	0.7529	1.2007	1.0514
38	2.0165	0.7528	1.1968	1.0488
39	2.0175	0.7531	1.2033	1.0533
40	2.0178	0.7533	1.1975	1.0494
41	2.0173	0.7531	1.2038	1.0536
42	1.9972	0.7456	1.2026	1.0503
43	2.0073	0.7493	1.2043	1.0527
44	2.0173	0.7531	1.2038	1.0536
45	2.0100	0.7503	1.2044	1.0531
46	1.9674	0.7345	1.2045	1.0478
47	1.9887	0.7424	1.2098	1.0540
48	2.0100	0.7503	1.2044	1.0531
49	1.9887	0.7424	1.2098	1.0540

Table 9. Ranking of Design Variable Combinations

Iteration 49 of optimization three was selected among all of the design choices for its excellent combination of volumetric and drag efficiency. This selection is shown graphically as Figure 80.

THIS PAGE INTENTIONALLY LEFT BLANK

V. PROTOTYPES

The objective of this study was to improve the fuel economy of light trucks and to quantify the improvement made possible through aerodynamic shape optimization. In order to confirm the possible improvements suggested by the CFD models it was necessary to develop full scale prototypes. The author selected some of the most promising models developed in the previous chapters to physically construct. This was done in order to prove that drag is a significant component of fuel consumption and to quantify the relative improvement made possible through aerodynamic optimization. As well, it is possible to qualitatively analyze the correlation of the drag predicted by the CFD models and the actual fuel consumption measurements.

The selection of the model to be prototyped on a full scale truck was detailed in the previous section. The model selected was the local minimum found in Optimization 3 because it scored the highest under the volumetric and drag combined selection criteria. This model is depicted below for comparison with what was actually constructed that will be seen in the subsequent sections.

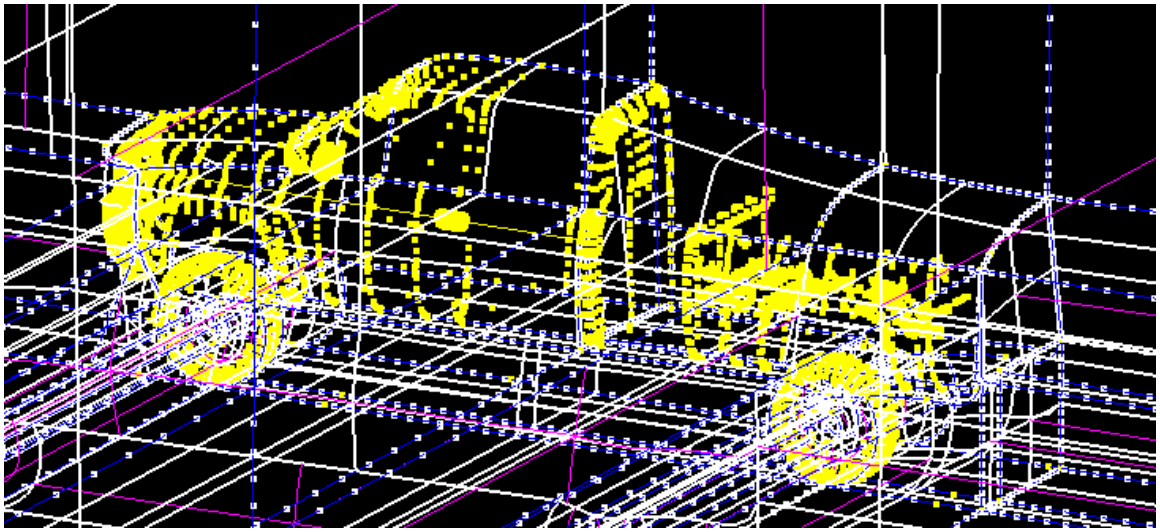

Figure 173. Optimization 3 Local Minimum and Design of Prototype

A. CONSTRUCTION OF THE AIR DAM

The air dam was found to be the most critical accessory component in the previous chapters. The initial design for the air dam called for a large rubber sheet that could flex and bend as the truck went over irregularities but would still be stiff enough to deflect the wind around the vehicle. These design criteria proved to be very difficult to accomplish in practice. The first several attempts by the author to build the air dam were largely unsuccessful because they were too flexible.

The initial design just consisted of hanging a rubber sheet from the front bumper. With only 4 inches of angle iron backing the rubber sheet, the design proved to be too flexible. The author next tried to extend the structure from the bumper with some plastic stiffening ribs that followed the rubber to the ground. This design also proved to be too flexible. It became evident that a substantial structure was necessary keep the rubber in place in order to actually function as an air dam.

Next the author abandoned the bumper/angle iron method and bolted a wooden structure directly to the frame of the truck with a bent plywood sheet to form a backing for the rubber. The plywood was designed to go within 6 inches of the ground. This provided enough stiffness when examined statically so that sheet could not be pushed back under the truck as before. Unfortunately this design also had a flaw in that it was only attached to the frame along one beam. The author immediately noticed excessive vibration and possibly an unsafe condition upon testing so a stiffer structure with more attachment points was deemed necessary.

The final working design [see fig. 174] combined elements of the previous ones. The author chose to bolt the plywood sheet to the bumper and maintain the current bolt pattern on the frame. Two more attachments were made to another beam of the under body and lastly Kevlar wire was used to hang the outermost structure from the outermost beam that had no bolt holes. Obviously this rudimentary design would not be acceptable for off road vehicles and a

welded metal design would be preferable. It should also be noted that the air dam was deployed with a 2 inch gap in order to avoid dragging. This was done realizing that the two inch gap is nearly as efficient as a sealed air dam as shown in Table 3. Recommendations for further design improvement are included in the last chapter.

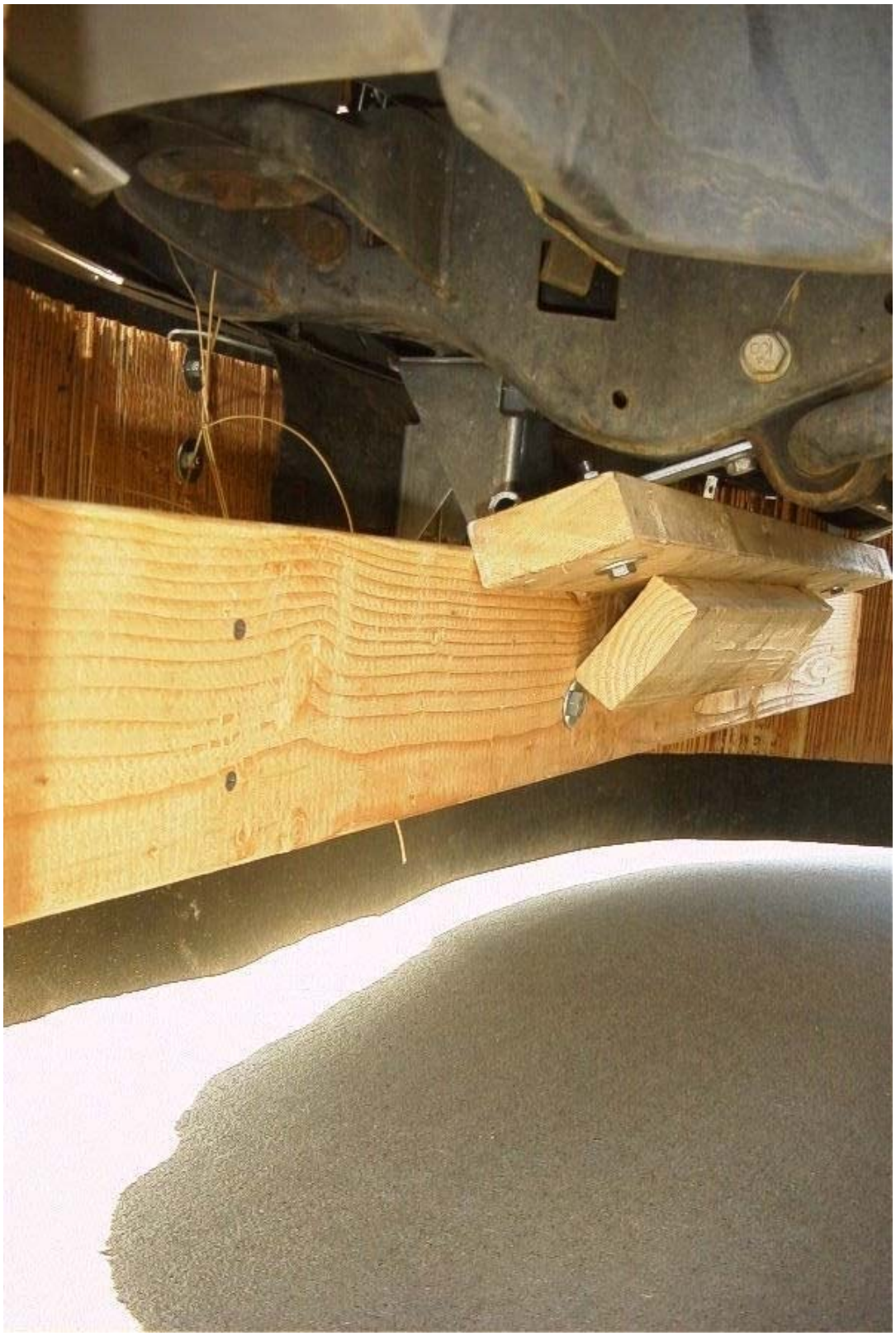

Figure 174. Air Dam Structure

Figure 175. Front View of Air Dam

Figure 176. Side View of Air Dam

B. CONSTRUCTION OF THE OPTIMAL CANOPY

The optimal canopy was constructed using wood, lath, and plaster. The first step [See fig. 177] was to build a very stiff plywood base that would not allow the plaster to flex as the vehicle was tested. The author ran two 2x6 and two 2x4 douglas fir beams longitudinally [See fig. 177] and one 2x6 across the bottom of the base (not visible in picture). This provided a very stiff base on which to build the superstructure. The next step was to transcribe the curves developed in CFD-GEOM, as a result of the optimization, to the plywood. The author took great care to plot the curves accurately on the plywood and to connect all points plotted with a very large parabolic curve. Additionally the author undersized the plywood curves by ½ inch to allow room for the layers of wood and plaster that were to come in the subsequent steps. The last element of the canopy structure was to stiffen the plywood curves by placing posts along the plywood curves and connecting them to the douglas fir beams.

Figure 177. Structure of Optimal Canopy

The next step was to run thin strips of wood longitudinally along the canopy in order to develop the parabolic shape created in CFD-GEOM. The strips easily conformed to the curvature of the design but provided ample resistance for the outer layers that were to follow.

Figure 178. Wooden Slats form Parabolic Shape

Next, metal lath [See fig. 179] was attached to the wood strips using drywall screws. This is the type of lath typically used by masonry building crews to finish the stucco exterior of a southwestern styled house.

Figure 179. Metal Lath Covers Wooden Slats

The third step in the process was to plaster the lath using paper backing and about 120 lbs of plaster. The plaster layer was made fairly thick (3/8 inch) in order to provide a significant surface for sanding. The plaster layer was very rough after the first application and took several days to dry in the sun due to its size [See fig. 180].

Figure 180. Paper then Plaster Applied to Lath

Lastly, the plaster model was planed and sanded. It was then placed upon the bed of the test vehicle. It should be readily evident in figures 181-183 that the plaster model, even when sanded, is not perfect and could benefit from additional skim coats and sanding to provide a smoother shape. As well, it would have been highly preferable to have made a fiberglass mold of this plug in order to provide a lighter and more realistic structure for testing. The air dam and canopy prototypes are heavier than production models would be by 300 to 400 lbs which should be considered in the results. Due to time constraints the author chose to test the prototypes at this point.

Figure 181. Rear Isometric View of Finished Optimal Canopy

Figure 182. Rear View of Finished Optimal Canopy

Figure 183. Side View of Finished Optimal Canopy

C. PROTOTYPE TEST METHODOLOGY AND RESULTS

The prototypes were tested in two configurations. The first configuration consisted of both the air dam and the optimal canopy on the vehicle. This was the local minimum arrived at in Optimization 3 of this thesis. The second configuration tested was with the air dam alone. It should be readily evident from tables two and three that the two inch gap air dam provides a sizable benefit on its own while the optimal canopy provides a negligible benefit on its own. As a result the author chose to pursue the air dam alone as the second test. Finally, the author also tested the vehicle without any accessories.

For all three test runs the same 101.5 mile route was used at as nearly the same speed as was possible. All three runs were conducted on the same day in the same conditions. The test run was approximately 75% freeway and 25% arterial/city driving. The author chose to test the vehicle at 70 mph on the freeway which is slightly below the average speed of the vehicles found on most of our nation's highways. The author was passed dozens of times by other

trucks and only overtook two or three vehicles on each run. This is anecdotal evidence that the 70mph run is approaching the actual driving characteristics of target population. This is in stark contrast to almost all of the data reported in the past where fuel economy has almost always been measured at 55 mph [Ref. 5,6]. The current EPA test, whose results are displayed on all new cars, never exceeds 60 mph [Ref, 5].

In order to provide as consistent of a test as possible the author topped of his gas tank before each run at the exact same fuel nozzle. The nozzle cutoff sensor is therefore assumed to not be a factor in the variation in fuel mileage. As well the author parked the vehicle at the nozzle with the wheels in the same position each time. This ensures that the vehicle was not angled differently in any direction to provide an erroneous fuel shut off in the nozzle. The same driver was used for all three runs and careful attention was paid to driving with the same acceleration characteristics each time. Lastly, for consistency the author timed each run in order to show that traffic and speed were relatively the same for all three runs.

The results of these runs and the predicted drop in drag by the CFD models are shown in the tables below:

Run	Comfiguration	Distance (miles)	Time (min)	Fuel (Gal)	Economy (mpg)
1	Baseline	101.5	105.5	5.31	19.11488
2	Air Dam Only	101.5	104.6	4.95	20.50505
3	Air Dam and Optimal Canopy	101.5	106.3	4.38	23.17352

Table 10. Fuel Economy Data

Run	Configuration	Mpg	% Difference in Fuel Economy Observed	% Difference in Drag Predicted
1	Baseline	19.11488		
2	Air Dam Only	20.50505	7.27%	16.72%
3	Air Dam and Optimal Canopy	23.17352	21.23%	37.62%

Table 11. Observed Fuel Economy Change and Predicted Change in Drag

It was expected that the observed fuel economy would not improve as much as the value of the drag prediction. This is obviously because the other losses in the system are unaffected by the change in aerodynamic resistance; such as: drive train losses and rolling resistance. Ultimately though the results were very encouraging and show that aerodynamics play a significant role in the characteristics of vehicles driven at modern speeds. As well, it has been shown that an optimized canopy and air dam provide a powerful combination in meeting the future fuel economy criteria.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

This thesis has quantified the relative merits of just under 250 light truck aero effect designs with respect to aerodynamics. Most of these designs were developed automatically during the optimization sequences yet a significant number were also developed to quantify existing accessory effects and some *ad hoc* theories. The *ad hoc* or intuitive research included the modeling of tonneau covers, air dams, side skirts, traditional covers, ducting and tailgate positions. The optimizations were focused solely on improving the shape of the canopy.

Both the intuitive and optimization methods of selecting designs proved to be very useful and predictive of possible improvements. Obviously the *ad hoc* approach of generating a design is necessary prior to optimization. This is the only way to test a theory and also provide a baseline for optimization. As a result it can be concluded that optimization greatly benefits from good early intuition on behalf of the analyst. This is especially true considering many of the 3D CFD models require three or more hours to run a single iteration.

Specific conclusions that can be drawn from this research are:

1. Two dimensional models do not accurately capture the physics involved with vehicle aerodynamics. They are very inaccurate when compared to their three dimensional counterparts when z direction flows are present.
2. There are multiple different ways to visualize the effects of drag. These include:
    a. Examining the pressure distribution resulting from a configuration for the effect of the pressure on the surface's normal vector
    b. Examining the velocity distribution for changes in momentum of the air particles
    c. Examining the pressure difference between the top and bottom of the truck for indications of induced drag.

d. While extended boundary layer attachment is often important it has to be evaluated within the context of the entire model. Clearly the boundary layer stayed attached for the entire length of the traditional canopy but this design did not fare as well as others (Figure 93). The turbulent air resulting from detachment is low pressure and must be evaluated in accordance with suggestion a. Essentially, a vacuum exists after the boundary layer detachment that pulls the vehicle against its direction of movement. If the normal of the area existing in this turbulent region is aligned with the direction of movement, greater drag occurs regardless of the extended boundary layer attachment. The traditional canopy exhibits this problem as its normal is perfectly aligned with the motion. The traditional canopy provides the worst results of all canopies (See Table 2). The converse is true for too early detachment of the boundary layer as in the tonneau cover (See Fig. 87). Figure 87 shows that very low pressure on the back of the cab results from too early of a separation. This is why the optimization strikes a perfect compromise between boundary layer separation and the orientation of the normal within the low pressure area.
e. Three dimensional CFD models predict drag and fuel economy well. See the prototype results on page 147.
f. Geometric drag optimizations are sizeable and provide a substantial benefit for light trucks.
g. Geometric changes do not always act independently. For example, the canopy only provides excellent results when used in conjunction with the air dam.
h. Intuition can be as important as optimization. This is evidenced by the benefits of the air dam versus all other changes. See Table 2 on page 85.

Ultimately, it was the synergy of a design arrived through intuition (the air dam) and an optimization (optimized canopy) that provided the best result. This

result was predicted to be very significant by the CFD solver and was on the order of a 40% reduction in drag. Previous research [Ref. 5,6] has indicated that approximately 50% of the fuel use is due to aerodynamic drag when traveling 55 mph. As a result, the author of this thesis expected that the fuel economy improvement due to reduced drag would be somewhat greater than 20% because the test was conducted at 70mph. This prediction was validated by the fuel economy results in the previous section. The prototype turned in a best fuel economy rating of a 21.23% improvement. Considering that there are 80 million light trucks on the road this would provide a substantial reduction in our nation's fuel consumption. A 20% improvement in all light trucks would result in saving 4 million of the 20 million barrels of oil used every day in the United States on light trucks.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. RECOMMENDATIONS

There are many areas that can be improved upon in this research or expanded into a new area. These areas of improvement can be broken down into two broad categories: error elimination and additional areas to investigate. First there were many sources of error in this research and future research would benefit greatly by reducing or eliminating such error. These errors and possible remedies are as follows:

1. The initial error came from the design blueprints on which the CFD models were based. Future researchers should endeavor to get exact CAD drawings from the manufacturer if not solid models. Solid models can be imported into a CFD program to automatically generate a very good grid. Thus future researchers should endeavor to create grids that are accurate to maximum extent possible.
2. The plaster model is very likely to riddled with geometric inconsistencies given the many layers and measurements necessary to develop it. Future research would benefit greatly by using a 3 axis computer numerically controlled (CNC) mill to develop any accessories or body panels. Much of the modeling error can be eliminated if this approach is combined with suggestion 1.
3. Develop accessories that would weigh the equivalent of a commercial system. In the case of this research, the air dam and plaster canopy weighed approximately 400 lbs. This is far more than a commercially developed system would weigh. In the future, body parts should be made out of carbon fiber or fiberglass to avoid this source of error. This probably resulted in poorer fuel economy that would have otherwise been observed.
4. Future air dams should be developed to raise and lower upon the input of the driver or a sensor. Future researchers should consider mounting an air dam on a hydraulically controlled snow plow linkage instead of merely bolting it to the frame as was done in this thesis.

This will allow for off-road capabilities and likely result in less air dam failures. In addition the air dam can be made to wrap much closer to the wheels and the ground for freeway use if a compromise between ground clearance and aerodynamics does not have to be made.

The second half of the recommendations is directed at possible new areas of study:

5. The effects of inclining the hood should be studied. The discontinuity in the transition from the hood to the windshield is suspected as a major source of drag. The hood and windshield are candidates for optimization.
6. Remove the mirrors to determine if they are significant sources of drag and can be replaced with closed circuit camera and LCD screens.
7. A height adjustable suspension could also provide a large benefit in reducing frontal area at high speeds.
8. Perform an optimization where the only constraints are dictated by general cabin, engine bay, and bed dimensions. Optimizing all surfaces would provide useful insight.
9. Find a way to incorporate the volume of a canopy into the cost function. This would allow for the goal programming of a given shape with a multiple objective cost function. This will require significant programming expertise in PYTHON and possibly the aid of CFDRC.

LIST OF REFERENCES

1. NASA http://www.grc.nasa.gov/WWW/K-12/airplane/drag1.html, Acessed: September 21, 2003
2. http://www.fhwa.dot.gov/ohim/onh00/line2.htm, Accessed: September 21, 2003
3. http://www.freep.com/money/autonews/cafe2_20030402.htm Accessed: September 21, 2003
4. E. J. Saltzman and R. R. Meyer, Jr. "A Reassessment of Heavy-Duty Truck Aerodynamic Design Features and Priorities", *NASA Report Number: NASA-TP-1999-206574*, 1999, pp 1-37.
5. J. Y. Wong, *Theory of Ground Vehicles (2nd Ed).* New York, NY. John Wiley and Sons, Inc. 1993.
6. F. M. White, *Fluid Mechanics (4th Ed).* Boston, MA. WCB McGraw-Hill, 1999.
7. H. K. Versteeg, and W Malalasekera, *An Introduction to Computational Fluid Dyanmics.* Essex, England. Longman Scientific and Technical, 1995.
8. G. Booch, *The Unified Modeling Language User Guide.* New York, NY. Addiso-Wesley, 1998.
9. J. A. Samareh. "A Survey of Shape Parameterization Technique" *CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics,* June 1999, pp. 333-343.
10. V. Braibant, and C. Fleury, "Shape Optimal Design Using B-Splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 44, No. 3,1984, pp. 247–267.
11. CFD Research Corporation, *CFD-ACE+^{TM} Simulation Manager: Script Based Simulation Reference Manual and Tutorials (Version 2002).* Huntsville, AL. 2002.

12. CFD Research Corporation, *CFD-GEOM™ Interactive Geometric Modeling and Grid Generation Software (Version 2002).* Huntsville, AL. 2002.
13. CFD Research Corporation, *CFD-ACE(U)™ User Manual (Version 2002).* Huntsville, AL. 2002.
14. A.D. Belegundu and T. R. Chandraupatla, *Optimization Concepts and Applications in Engineering.* Upper Saddle River, NJ. Prentice Hall, Inc., 1999.

APPENDIX

```
1. # This file was generated by CFD-GEOM
2. import GPoint
3. import GCurve
4. import GSurface
5. import GLoop
6. import GEdge
7. import GFace
8. import GBlock2D
9. import GBlock
10. import GEntity
11. import GManip
12. import GUnstruct
13. import GInterface
14.
15.
16. # Set Geometric Precision
17. GEntity.SetPrecision ( 1E-006 )
18.
19. GInterface.GGDRead ('C:/Documents and Settings/nawillia/My
    Documents/Thesis/3DTruckFiles/3Dtradrecover2.GGD')
20. GEntity.Delete (GEntity.GetEntityByName ('geom_block29'))
21. GEntity.Delete (GEntity.GetEntityByName
    ('geom_op_ext4[\'blocks\'][3]'))
22. GEntity.Delete (GEntity.GetEntityByName ('geom_face468'))
23. GEntity.Delete (GEntity.GetEntityByName ('geom_face467'))
24. GEntity.Delete (GEntity.GetEntityByName
    ('geom_op_trs11[\'edges\'][193]'))
25. GEntity.Delete (GEntity.GetEntityByName
    ('geom_op_trs11[\'curves\'][256]'))
26. backmove = -.250489
27. GEntity.CreateParameter (globals (), 'backmove')
28. geom_op_trs12 = GManip.Translate ([0, 1, 0], backmove,
    GEntity.GetEntityByName ('geom_op_trs11[\'points\'][6192]'))
29. backmove2 = -.75
30. GEntity.CreateParameter (globals (), 'backmove2')
31. geom_op_trs13 = GManip.Translate ([0, 1, 0], backmove2,
    GEntity.GetEntityByName
    ('geom_op_trs11[\'points\'][6173][\'points\'][0]'))
32. midedgey = 1.64608
33. GEntity.CreateParameter (globals (), 'midedgey')
34. geom_point6208 = GPoint.Create (-17.5, midedgey, .8)
35. midcentery = 1.74108
36. GEntity.CreateParameter (globals (), 'midcentery')
```

37. geom_point6209 = GPoint.Create (-17.5, midcentery, 0)
38. geom_curve789 = GCurve.CreateThroughPoints ([geom_point6209,geom_point6208,GEntity.GetEntityByName ('geom_op_trs11[\'points\'][6059]')])
39. geom_curve790 = GCurve.CreateThroughPoints ([GEntity.GetEntityByName ('geom_op_trs11[\'points\'][93]'),geom_point6209,GEntity.GetEntityByName ('geom_op_trs11[\'points\'][6192]')])
40. geom_op_spl1 = GCurve.SplitAtPoint (geom_curve790, geom_point6209)
41. geom_edge641 = GEdge.CreateFromCurves (geom_op_spl1['curves'][1], 18, 1, 1, -1.0, -1.0, 0, -1.0, -1.0)
42. geom_edge642 = GEdge.CreateFromCurves (geom_op_spl1['curves'][0], 18, 1, 1, -1.0, -1.0, 0, -1.0, -1.0)
43. geom_edge643 = GEdge.CreateFromCurves (geom_curve789, 19, 1, 1, -1.0, -1.0, 0, -1.0, -1.0)
44. geom_face474 = GFace.Create (geom_edge641,geom_edge643,GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][10]'),GEntity.GetEntityByName ('geom_op_trs11[\'points\'][6173][\'edges\'][0]'))
45. geom_face475 = GFace.Create (geom_edge642,geom_edge643,GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][15]'),GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][17]'))
46. geom_face476 = GFace.Create (GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][214]'),[geom_edge642,geom_edge641],GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][94]')
47. GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][95]'))
48. geom_block31 = GBlock.Create ([GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][49]'),GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][88]')],geom_face476
49. GEntity.GetEntityByName ('geom_face470'),[GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][50]'),GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][87]')],[geom_face474,geom_face475
50. GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][85]'),GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][86]')],GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][84]'))
51. geom_block32 = GBlock.Create ([GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][15]'),GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][16]')],GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][14]')
52. GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][13]'),GEntity.GetEntityByName

```
('geom_op_ext4[\'faces\'][17]'),[GEntity.GetEntityByName
('geom_op_ext4[\'faces\'][18]')
GEntity.GetEntityByName
('geom_op_ext4[\'faces\'][11]')],GEntity.GetEntityByName
('geom_face470'))
GEntity.SetBC (geom_face476, 'Symmetry')

GEntity.Delete                    (GEntity.GetEntityByName
('geom_op_trs11[\'blocks\'][9]'))



GEntity.Delete                    (GEntity.GetEntityByName
('geom_op_trs11[\'blocks\'][2]'))
GEntity.Delete                    (GEntity.GetEntityByName
('geom_op_trs11[\'blocks\'][7]'))
GEntity.Delete                    (GEntity.GetEntityByName
('geom_op_trs11[\'blocks\'][5]'))
GEntity.Delete                    (GEntity.GetEntityByName
('geom_op_trs11[\'blocks\'][3]'))


GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
s\'][3]'), 'cabtop')
GEntity.SetBCName (geom_face475, 'canopy1')
GEntity.SetBCName (geom_face474, 'canopy2')
GEntity.SetBCName(GEntity.GetEntityByName('geom_face471'),
'back1')
GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
s\'][125]'), 'back2')
GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
s\'][124]'), 'back3')
GEntity.SetBCName(GEntity.GetEntityByName
('geom_op_trs11[\'faces\'][8]'), 'rearquarter')
GEntity.SetBCName(GEntity.GetEntityByName
('geom_op_trs11[\'faces\'][6]'), 'rearwheel')
GEntity.SetBCName(GEntity.GetEntityByName
('geom_op_trs11[\'faces\'][56]'), 'rearhub')
GEntity.SetBCName
(GEntity.GetEntityByName('geom_face473'),'rearside')
GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
s\'][43]'), 'skirt')
GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
s\'][42]'), 'frontside')
GEntity.SetBCName(GEntity.GetEntityByName
('geom_op_trs11[\'faces\'][7]'), 'frontwheel')
```

```
79. GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
    s\'][57]'), 'fronthub')
80. GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
    s\'][62]'), 'damside')
81. GEntity.SetBCName(GEntity.GetEntityByName
    ('geom_op_trs11[\'faces\'][64]'), 'damfront')
82. GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
    s\'][65]'), 'nose')
83. GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
    s\'][2]'), 'hood')
84. GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
    s\'][4]'), 'windshield')
85.
86.
87.
88.
89.
90.
91. GEntity.Delete (geom_block32)
92. GEntity.Delete (geom_block31)
93. GEntity.SetBCName(GEntity.GetEntityByName('geom_op_trs11[\'face
    s\'][121]'), 'canopy1')
94. GEntity.Delete(GEntity.GetEntityByName('geom_op_ext4[\'blocks\'][2]')
    )
95. GEntity.Delete (GEntity.GetEntityByName ('geom_face471'))
96. GEntity.Delete (geom_face474)
97. GEntity.Delete(GEntity.GetEntityByName('geom_op_ext4[\'faces\'][11]')
    )
98. GEntity.Delete (GEntity.GetEntityByName ('geom_face470'))
99. GEntity.Delete(GEntity.GetEntityByName('geom_op_trs11[\'points\'][61
    73][\'edges\'][0]'))
100.    GEntity.Delete(GEntity.GetEntityByName('geom_op_trs11[\'points\']
    [6173][\'curves\'][0]'))
101.    GEntity.Delete(GEntity.GetEntityByName('geom_op_trs11[\'points\'[
    6173][\'curves\'][1]'))
102.    GEntity.Delete(GEntity.GetEntityByName('geom_op_trs11[\'points\']
    [6173][\'points\'][0]'))
103.    backedge = 1.57001
104.    GEntity.CreateParameter (globals (), 'backedge')
105.    geom_point6210 = GPoint.Create (-16.411, backedge, .72091)
106.    geom_curve793
    =GCurve.CreateThroughPoints([GEntity.GetEntityByName('geom_op_t
    rs11[\'points\'][6173]'),geom_point6210,GEntity.GetEntityByName('geo
    m_op_trs11[\'points\'][6192]')])
107.    geom_edge644 = GEdge.CreateFromCurves (geom_curve793, 19,
    1, 1, -1.0, -1.0, 0, -1.0, -1.0)
```

108. geom_face477 = GFace.Create (geom_edge644,GEntity.GetEntityByName ('geom_edge622'),[GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][199]')
109. GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][196]')],GEntity.GetEntityByName ('geom_edge640'))
110. geom_face478 = GFace.Create (geom_edge641,geom_edge643,geom_edge644,GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][10]'))
111. geom_face479 = GFace.Create (geom_edge644,GEntity.GetEntityByName ('geom_op_ext4[\'edges\'][17]'),GEntity.GetEntityByName ('geom_op_ext4[\'edges\'][14]')
112. GEntity.GetEntityByName ('geom_op_ext4[\'edges\'][16]'))
113. geom_face480 = GFace.Create (GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][214]'),[GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][202]')
114. GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][211]')],GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][209]'),[GEntity.GetEntityByName ('geom_op_trs11[\'edges\'][210]'),geom_edge644])
115. geom_block33 = GBlock.Create ([GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][18]'),geom_face479],GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][14]')
116. [GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][15]'),GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][16]')],GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][17]')
117. GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][13]'),geom_face480)
118. geom_block34 = GBlock.Create (geom_face477,GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][12]'),geom_face479,GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][10]')
119. [GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][1]'),GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][6]')],GEntity.GetEntityByName ('geom_op_ext4[\'faces\'][9]'))
120. geom_block35 = GBlock.Create (geom_face476,[geom_face478,geom_face475,GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][85]'),GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][86]')]
121. GEntity.GetEntityByName ('geom_op_trs11[\'faces\'][84]'),[GEntity.GetEntityByName

```
    ('geom_op_trs11[\'faces\'][88]'),GEntity.GetEntityByName
    ('geom_op_trs11[\'faces\'][49]')],geom_face480
122.  [GEntity.GetEntityByName
    ('geom_op_trs11[\'faces\'][87]'),GEntity.GetEntityByName
    ('geom_op_trs11[\'faces\'][50]')])
123.  GEntity.SetBCName (geom_face477, 'back1')
124.  GEntity.SetBCName (geom_face478, 'canopy2')
125.
126.  GEntity.SetBCName                    (GEntity.GetEntityByName
    ('geom_op_trs11[\'faces\'][121]'), 'bottom')
127.  GInterface.DTFWrite_3d ('3DOpttimal.DTF')
```

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Mike Leatherwood

   US Army Tank-automotive and Armaments Command
   6501 E. Eleven Mile Road
   Warren, Michigan
   48397-5000
   Attention: Mr. Mike Letherwood
   AMSTA-TR-N, MS #157

4. D.C. Boger
   Naval Postgraduate School
   Monterey, California

5. J.H. Gordis
   Naval Postgraduate School
   Monterey, California